

Jak pisał nieodzałowany Tolkien: "Świat sie J.R.R. zmienił. Mówi mi to woda. Mówi mi to ziemia. Wyczuwam to w powietrzu. Wiele z tego, co było, zostało zatracone, bo nie żyją już Ci, którzy to pamiętali...". Z pewnością nie chcemy, aby taki los był udziałem naszych ulubionych komputerów, dlatego staramy się propagować wiedzę o ich budowie, działaniu i możliwościach. Wraz z tym trendem zmienia sie nasz magazyn i mam nadzieję, że moje propozycje przypadną Wam do gustu.

Po pierwsze: zmniejszamy puste przestrzenie oraz czcionki na wszystkich stronach. Nie jest to to bardzo widoczne, ale odczuwalna jest zwiększona ilość tekstu, szczególnie podczas korekty. Bez zwiększenia objętości możemy teraz przekazać Wam większą ilość informacji, co mam nadzieję zaowocuje również uzyskaniem większej równowagi pomiędzy poszczególnymi działami. Ponadto, w wielu wypadkach będziemy zmniejszać ilustracje, choć nie zawsze jest to celowe, dlatego nadal najwyższy priorytet ma ogólny układ stron.

Po drugie: ruszamy z nowym działem "IBM PC". Będzie to miejsce do prezentacji różnych pomysłów na używanie retro pecetów, podobnie jak w strefie Apple. Nie będziemy oczywiście pisać o nowoczesnych modelach PC, a nowy dział pozostanie mały w stosunku do całej reszty magazynu. Nie musicie obawiać się, że chcemy zmienić RetroKompa w "retro PC", a takie komentarze usłyszałem na ostatniej imprezie w Gdańsku. Chcemy pokazać, że starego peceta można używać z takim samym (lub prawie) zadowoleniem jak Commodore czy Atari, a w wielu wypadkach sposób użycia może być wręcz analogiczny. Trzeba tylko oduczyć się traktowania tej platformy przez pryzmat duetu Microsoftu z systemem Windows.

Jeśli już jesteśmy przy tym temacie, chciałbym bardzo serdecznie podzuczestnikom iękować wszystkim RetroKomp 2018. Pokazaliście, że ciagle interesujecie się platformami retro i dotyczy to nie tylko starych wyjadaczy, lecz również ludzi dopiero wchodzących w swoje dorosłe życie. Gracze moga być zawiedzeni mniejsza ilościa produkcji pokazanych w konkursie GameDev, ale moim zdaniem ich jakość to rekompensuje. Rzecz jasna, nadal nie możemy oczekiwać poziomu gier z okresu największego komercyjnego rozkwitu choćby Amigi, ale widać, że programiści "idą do przodu" i dopracowują nie tylko pomysły, ale także swój warsztat.

że wchodzimy w Myślę, ilowog przełomowy okres dla naszego środowiska. Mianowicie, niektórzy wycofują się ze świata retro ze względu na zniechęcenie lub zmiany życiowe. Inni odkrywają sprzęt, do którego nie mieli kiedyś dostępu i okazuje się, że sentyment to nie wszystko. Jednak największa grupa fanów umacnia się w przekonaniach i "rozpracowuje" swoje komputery pod względem sprzętowym lub programowym, co z pewnością będzie owocowało nowym oprogramowaniem, choć pewnie w większości będą to dema i proste gry.

Świat się zmienia, my się zmieniamy, ale nie muszą zmieniać się nasze zainteresowania. Ważne, aby znaleźć czas na swoje ulubione hobby i pokazać ludziom wokół, że nie liczy się tylko to, co akurat jest na topie. Być może uda się Wam w ten sposób zainteresować tematem retro kolejne osoby, do tej pory przekonane o niekwestionowanej roli peceta na rynku komputerów. I tego życzę Wam na nadchodzącą Gwiazdkę. Kolejny numer ukaże sie dopiero po Nowym Roku, w którym mam nadzieję nadal będziecie entuzjastów zasilać liczną grupę prawdziwych komputerów z duszą.

Adam Zalepa



NUMER 12/2018

ISSN 2450-5862 Redaktor naczelny: Adam Zalepa

Autorzy tekstów: Mateusz "Tfardy" Eckert Krzysztof Kliś Piotr "Piter" Krużycki Marcin Libicki Piotr "Kroll" Mietniowski Michał "stRing" Radecki-Mikulicz Leszek Rosiński Piotr "Sachy" Sachanowicz Marcin Skawiński Kamil Stokowski Mariusz Wasilewski

> Projekt okładki: Marzena Bukowska

Korekta: Renata Gralak Sklad: Andrzej Wilczyński

Wydawca: A2 Renata Gralak, Łódź

Wszystkie nazwy oraz znaki handlowe należą do ich właścicieli i zostały użyte wyłącznie w celach informacyjnych.



RETROKOMP 12:

Świat się zmienia	
INTEGRACJA	5
Fujitsu Micro 16s TRS-80 Touch Pad Epson HX-20 IBM PC Jr Sega Teradrive Wizyta w gdańskim muzeum RetroKomp Język Logo w polskich szkołach	
ZX SPECTRUM	18
Procedury dźwiękowe Pierwszy program w języku maszynowym	
AMSTRAD	23
Uruchamiamy system FutureOS	
COMMODORE	25
Remont Commodore 64 Instrukcja USR w przykładach Kontrola czau na VIC-20 Testowanie dźwięku SID	
ATARI	35
Atari Light Pen Rysowanie światłem – zalety i wady Kocie nowości Bufor klawiatury Wykopaliska	
APPLE	46
Większy dysk w systemie DOS 3.3 Menu rozwijane dla Apple II Macintosh kontra MS-DOS – część 2. Globalne zapisywanie zmiennych	
ACORN	59
Archimedes A3010 Programowanie s systemie RISC OS – część 1 Kolory w Basicu	
AMIGA	64
Grafika w formacie PNG Kompresja danych na procesorach PowerPC	
IBM PC	72
Fraktale i inne figury Pliki wsadowe w IBM Basic PC Jr i program w jednej linii Quadscreen	

Fujitsu Micro 16s

Komputer o nazwie Micro 16s jest bardzo ciekawym rozwiązaniem, bowiem wyposażony go w dwa mikroprocesory. Jednym z nich jest 8bitowy Z80A, a drugim - już 16-bitowy Intel 8086. Poza tym posiada 128 kilobajty pamięci RAM, dwa napędy dyskietek 5,25-cala i odłączaną klawiaturę. Wszystko działa pod kontrolą słynnego systemu CP/M.

Według Fujitsu, Micro 16s został zaprojektowany na rynek amerykański, w szczególności dla użytkowników biznesowych. Został wprowadzony na ry nek w 1983 roku. Kluczowymi cechami są dwa mikroprocesory i system operacyjny, pod którym procesor Z80 może wykonywać oprogramowanie zaprojektowane dla CP/M w wersji 2.2. Uruchomienie opcjonalnego kodu dla procesora Intela pozwala wykonywać jednocześnie do czterech zadań obliczeniowych.

Micro 16s zawiera 128 kilobajtów pamięci. z kontrolą parzystości, którą można rozszerzyć do jednego megabajta. Ponadto maszyna posiada dodatkowe 48 kilobajty przeznaczone do obsługi obrazu w wysokiej rozdzielczości. Nie mniej ważne jest dołączone oprogramowanie, wśród którego należy wymienić takie pozycje jak WordStar i SuperCalc. Fujitsu w tym wypadku zdecydowało się nie udostępniać języków programowania wysokiego poziomu, przez co swoją politykę zbliżyło do dzisiejszego podejścia do komputerów używanych w domu.

Również wiele dodatkowych programów było sprzedawanych nie przez Fujitsu, lecz przez lokalnych dystrybutorów. Tłumaczono to filozofią marketingową polegającą na skierowaniu Micro 16s do klienta biznesowego poszukującego gotowego systemu "pod klucz".

Oczywiście nie oznacza to, że tym komputerze nie można programować. Wręcz przeciwnie, podręcznik do systemu CP/M zawiera dokumentacje odnoszącą się do Asemblera. Jednak nie jest to ten rodzaj szczegółowości, do którego przyzwyczaili nas producenci sprzętu w latach '80-tych, instrukcja iest bardziei oqólnvm omówieniem możliwości wraz wieloma odniesieniami do dokumentacji procesora 8086. Instrukcja zawiera za to obszerne instrukcje dotyczące rozpakowywania, konfigurowania i uruchamiania komputera.

Obudowa Micro 16s jest metalowa, nieco mniejsza niż ówczesnych produktów firmy IBM. Posiada osobne złącza dla klawiatury, pióra świetlnego, monitora monochromatycznego i kolorowego RGB, drukarki, a także urządzeń podłączanych do portu szeregowego RS-232. Mamy także dostęp do specjalnego panelu zawierającego 16 przełączników typu DIP. Nie są one opisane w instrukcji, z wyjątkiem jednej ilustracji pokazującej, w jaki sposób powinny być ustawione w celu prawidłowego działania monitora.

Z tyłu jednostki znajdują się cztery prostokątne osłony, które kryją 5 gniazd rozszerzeń, każde 130-pinowe. Nie przypomina to żadnego typowego portu, na dodatek w instrukcji nie ma na żadnych ten temat informacji. Użytkownik, który kupował wtedy Micro 16s mógł najwyżej snuć przypuszczenia, jaki dodatkowy osprzęt będzie mógł podłączyć do swojego komputera za jakiś, bliżej nieokreślony, czas.

Komputer ma także dwie stacje dyskietek 5,25-cala, obsługującej dwustronne nośniki o pojemności 320 kilobajtów. Dioda LED pokazuje, kiedy są wykonywane operacje. Ciekawym rozwiązaniem jest wydłużenie czasu obracania się napędu po zakończeniu odczytu lub zapisu, co przyspiesza dostęp do danych. Przypomina to nieco rozwiązania stosowane w niektórych drukarkach laserowych, na przykład HP LaserJet 1160.

Klawiatura posiada 98 klawiszy, w tym standardowa klawiaturę alfanumeryczną, a także klawiaturę numeryczną operacjami arytmety-Ζ cznymi, dziesięć klawiszy funkcyjnych, dziesięć klawiszy specjalnych i cztery sterowania klawisze kursorem ułożone w logicznie uzasadniony sposób, choć wygląda to nietypowo. Miłym akcentem jest dioda LED obok klawisza CAPS LOCK i podobny wskaźnik dla klawisza INSERT. Klawisz ENTER jest ok. czterokrotnie większy od pozostałych klawiszy.

Klawisz GRAPH służy do uzyskiwania znaków graficznych z klawiatury. Jednak nie jest to klawisz przełączający tryb pracy, lecz musi być przytrzymany razem z innym klawiszem. Działa to podobnie do kombinacji klawiszy typu ALT czy CTRL i powoduje wprowadzenie alternatywnego zestawu znaków. W sumie klawiatura jest w stanie wygenerować 96 standardowych liter, cyfr i symboli, 32 znaki graficzne, 39 liter greckich, 10 liczba o zmniejszonych rozmiarów i 26 symboli matematycznych, muzycznych symbole i naukowych.

Klawiatura ma również kilka interesujących stałych klawiszy funkcyjnych, w tym wstawianie, usuwanie, kasowanie linii, kasowanie ekranu, ustawianie kursora w pozycji "domowej" i duplikowanie znaków. Ostatnia funkcja działa inaczej w różnych pakietach oprogramowania, ale ogólne przeznaczenie jest zachowane we wszystkich dołączonych programach.

Za pomocą polecenia KLICK można włączać i wyłączać dźwięki klawiszy, oczywiście dotyczy to systemu CP/M. Powoduje to ciche "kliknięcia", które działają inaczej w różnych programach. Wszystkie klawisze posiadają samopowtarzalność po przytrzymaniu przez ok. 1 sekundę. Standardowe klawisze alfabetyczne, numeryczne i symbole są białe z czarnymi napisami, a pozostałe



Komputer Fujitsu 16s stylistycznie przypomina produkty IBM.

klawisze są jasno-szare. Ten sam schemat kolorów jest również używany na obudowie.

W sprzedaży dostępne były dwa monitory firmy Fujitsu: Micro 16s, 12-calowy monochromatyczny (zielony) i 11calowy kolorowy (RGB). Rozdzielczość w trybie tekstowym wynosi 40 lub 80 znaków na 25 linii. Za pomocą polecenia SCREEN w CP/M można zmieniać ustawienia poprzez definiowanie różnych rodzajów zmiennych. Modyfikować możemy między innymi liczbę linii na ekranie, liczbę przewijanych automatycznie linii, ilość wyświetlanych kolorów oraz tryb wyświetlania zegara czasu rzeczywistego.

Rozdzielczość grafiki wynosi 640x200 pikseli. Z kolorowym monitorem dostępnych jest osiem kolorów. Pamięć wideo dla grafiki i trybu znakowego jest oddzielna, więc mogą być wyświetlane niezależnie. Różne pakiety oprogramowania wykorzystują kolory na różne sposoby. Na przykład SuperCalc używa białego jako hasła użytkownika, niebieskiego dla nazw w arkuszach kalkulacyjnych i jasno-niebieskiego dla etykiet klawiszy funkcyjnych.

Do monitora można było też dokupić obrotowy uchwyt, który podnosi nieco monitor i pozwala go przechylać od ok. 5 stopni do przodu do 20 stopni oraz obracać o 45 stopni w dowolną stronę.

Gdy system operacyjny jest uruchamiany po raz pierwszy, czyli bez dysku w napędzie, na ekranie pojawia się następujący komunikat:

DISK ERROR OA (R, H, D, S, O, X or G)

Instrukcja mówi, że jest to normalne, ciekawe jednak, co oznaczają wyświetlone litery. Fujistu, zgodnie ze swoją filozofią "upraszczania" obsługi, nie uznał jako celowe poinformować użytkownika o stosowanych symbolach,

dlatego niezależnie stopnia od szczegółowości tego komunikatu pozostaje po prostu włożyć dyskietkę do napędu i zresetować komputer. Szkoda, że oryginalna dokumentacja jest tak skromna, ale myślę, że producent nie sądził, że odbiorca docelowy będzie chciał tak szczegółowo zajmować się prostymi komunikatami o błędach, szczególnie po wielu latach.

CP/M-86 to system operacyjny zaprojektowany przez firmę Digital Research i w tym wypadku działa pod kontrolą procesora Intela. Jesto on zgodny ze standardowym CP/M dla Z80, co powoduje, że bez problemu odczytami dyskietki - o ile do zapisu stosowany był domyślny system plikowy.

Można powiedzieć, że jest to most pomiędzy oprogramowaniem dla 8- i 16-bitowych procesorów, a także możliwość łatwego przenoszenia danych z innych komputerów. CP/M-w tej wersji może obsługiwać do jednego megabajta wewnętrznej pamięci RAM, pracować na 16 dyskach logicznych o pojemności do ośmiu megabajtów i kilku dodatkowych urządzeń.

Micro 16s ma dwa procesory, dlatego programy nie muszą być w żaden sposób dostosowywane. Są po prostu uruchamiane bezpośrednio korzystając z właściwego procesora. Niestety nie odbywa się to automatycznie, ale trzeba najpierw wczytać system w wersji dla procesora 8086, a następnie uruchomić specjalny program obsługujący CP/M w wersji 2.2.

Dodatkowym ograniczeniem jest fakt, że programy nie mogą korzystać z odwołań sprzętowych, ani sztuczek związanych z konkretną architekturą. Powoduje to, że część programów wyświetla nieprzewidziane błędy, jak na przykład Perfect Calc czy Perfect Writer, które wywołują komunikaty typu "disk not ready:" lub "I / O error". Warto dodać kilka słów o kilku dołączonych programach, które w latach '80tych były znane, a dzisiaj raczej należą do zapomnianej historii. Pierwszym z nich jest WordStar, czyli najczęściej stosowany ówcześnie procesor tekstu. Był uznawany za wyjątkowo wszechstronny, posiadał wiele niestandardowych funkcji i udoskonaleń, także w specjalnej wersji dla Micro 16s. Do programu użytkownik otrzymywać ponad 200 stronicowy podręcznik, omawiający instalację oraz obsługę, wraz z funkcjami niestandardowymi. Usprawnienia dotyczyły między innymi lepszej jakości druku na drukarkach Epson.

Kolejnym ważnym programem jest SuperCalc, czyli pakiet arkusza kalkulacyjnego dostosowanego do odbiorcy biznesowego. Jako nowe funkcje wymieniano wtedy automatyczne sor-

press	ACP/M AssemblerNMailmergeBCBASICOOsborne UtilitiesCCP/M CommandsPPrinters/PrintingDDiskette HandlingQQuitting Each BayEFile ExtensionsRThe RESET ButtonFFilenamesSSuperCalcGGraphicsTTestingHProgramming HintsUCP/M UtilitiesJJust Starting?WWordStarKControl KeysXRccessoriesLLayout of MemoryYModen ConnectionMMoresoft BRSICZSelf Portrait

Przykładowe ekranu systemu operacyjnego CP/M.





Reklama prasowa producenta systemu CP/M.

towanie wierszy lub kolumn, formatowanie tekstu oraz liczb, zabezpieczanie komórek lub całych zakresów komórek, funkcje logiczne oraz funkcje związane z operacjami na różnych formatach daty. SuperCalc w wersji dla komputera Fujitsu jest także przystosowany do korzystania z dziesięciu wbudowanych i dwóch programowalnych klawiszy funkcyjnych, a także pozwala zmieniać paletę kolorów na ekranie.

Specyfikacja Micro 16s wymienia kilka elementów, które trudno sprawdzić dzisiaj, lecz mówi się o możliwości podłączania zewnętrznych twardych dysków oraz 8-calowych stacji dyskietek. Dodatkowo, sprzęt może pracować w sieci Omninet, która została opracowana przez firmę Corvus. Dzięki temu system może pracować jako węzeł większej sieci, dzieląc zasoby i programy z innymi maszynami.

Tego typu sieci były stosowane od początku lat '80-tych i można powiedzieć, że był to odpowiednik dzisiejszej sieci lokalnej ty- pu Ethernet.

Wspomniane rozszerzenia były prawdopodobnie przewidziane do podłączania do 130-pinowych portów, o których pisałem wcześniej.

Kolejną ciekawostką jest fakt, iż prasa omawiając Micro 16s wskazywała często brak możliwości uruchomienia Microsoft Basica jako jeden z największych błędów producenta. Sprzęt był oceniany jako bardzo wydajny, lecz także trudny w obsłudze.

Z pewnością był to pogląd niezbyt trafny, ale mówiono tak patrząc przez pryzmat systemu operacyjnego CP/M. Basic wydawał się

bardziej naturalny i lepiej przystosowany do przeciętnego użytkownika. Z drugiej strony komputer Fujitsu był skierowany na rynek biznesowy, tak więc starano się sprawić, aby prezentował się jak najbardziej profesjonalnie, nawet jeśli w części były to tylko pozory.

Dla mnie historia Micro 16s pokazuje, jak bardzie zmieniło się podejście użytkownika do komputera osobistego. Dziś możliwość wprowadzania programów w Basicu lub innym języku wysokiego poziomu oceniamy jako czynność specjalistyczną i w gruncie rzeczy mało kto chce się tym zajmować. Osobiście ubolewam, że minely czasy, gdy komputer był traktowany jako narzędzie specjalistyczne, a użytkownika jako osoby, która musi nabyć minimalną wiedzę z zakresu programowania, aby sprawnie obsługiwać system operacyjny.

> Opracował: Marcin Libicki

Nowe wydawnictwa dla każdej platformy sprzętowej





TRS-80 Touch Pad

Komputery TRS-80 nie były w Polsce nigdy szeroko znane i popularne, a swoje sukcesy święciły w USA od początku lat '80-tych. Tandy, wyprodukowała wiele ciekawych akcesoriów, często nietypowych jak na dodatek do sprzętu 8bitowego. Jednym z nich jest Touch Pad.

Zanim jednak do niego dojdziemy, przedstawię w skrócie historię i architekturę sprzętu o nazwie TRS-80 Micro Computer System. Model TRS-80, który został później przemianowany na "Model I" w celu odróżnienia go od następców, jest mikrokomputerem wprowadzonym na rynek jeszcze w 1977 roku.

Był sprzedawany przez Tandy Corporation za pośrednictwem sieci swoich sklepów o nazwie Radio Shack. Był to jeden z najwcześniej produkowanych i masowo sprzedawanych komputerów domowych w czasach. gdy nie było jeszcze ZX Spectrum ani Commodore 64.

Model TRS-80 wyposażono w pełną klawiaturę QWERTY, procesor Zilog Z80, zamiast bardziej popularnego Intel 8080, standardową pamięć DRAM 4 KB, co było duża ilością, bowiem wiele komputerów 8-bitowych dostarczanych było z tylko 1 KB pamięci RAM. Maszyna ma małe rozmiary i posiada wbudowany interpreter języka programowania Basic i standardowy 64znakowy monitor.

W późniejszym czasie producent opracował rozszerzenia pozwalające między innymi zwiększyć pamięć do 48 KB, a także podłączyć twardy dysku lub nawet 4 stacje dyskietek. Jednym z ciekawszych dodatków był także TRS-

80 TouchPad. Jest to panel dotykowy, który posiada powierz- chnię wrażliwą na nacisk, tak więc można go używać zarówno za pomocą dołączonego ry- sika, jak i palca.

Tablet posiad przyciski sterujące, których można używać do potwierdzania wyborów opcji menu, wprowadzania danych lub, co ciekawe, jako kontrolery grach. w Rysik został specjalne zaprojektowany przy uwzględneniu powierzchni dotykowej, tak więc cały zestaw został stworzony jako całość, w profesjonalny sposób.

Jednak, gdy przyjrzymy się bliżej historii TRS-80 TouchPada, okaże się, że był to produkt sprzedawany wcześniej przez amerykańską firmę Koala Technologiers pod nazwą KoalaPad. Był on przeznaczony dla kilku 8-bitowych komputerów domowych, w tym oczywiście rodziny Apple II, TRS-80 Color, 8-bitowego Atari, Commodore 64, a także IBM PC.

Tablet był pierwotnie zaprojektowany jako tanie narzędzie do rysowania z



przeznaczeniem dla szkół. Dołączano do niego program do rysowania o nazwie KoalaPainter, bardzo popularny wśród bvł użytkowników domowych. Dodajmy, że w niektórych wersjach, na przykład dla Apple II, ten sam program był sprzedawany jako Koala-Paint, natomiast w wersji dla PC nazwano go "zawodowo" - PC Design.

INTEGRACIA

Powierzchnia tabletu do 4-calowy kwadrat (czyli ok. 10x10 cm) i jest zamontowana na lekko pochylonej podstawie. Na górze, czyli "z tyłu" umieszczone są wspomniane już dwa przyciski. Całość należy podłączyć do komputera za pomocą portów joysticka, co oznacza, że urządzenie siłą rzeczy posiada dość niską rozdzielczość.

Z dzisiejszego punktu widzenia jest to głównie ciekawostka, a przyciski są toporne i nieco denerwujące w użyciu. Jednak pamiętajmy, że jest to produkt sprzed prawie 40 lat i wprowadza do świata komputerów 8-bitowych tak popularną dzisiaj dotykową filozofię pracy.

Oczywiście nie było możliwe zastosowanie ekranu dotykowego, ale praca na TouchPadzie jest zdecydowanie wygodniejsza niż przesuwanie emulowanego kursora za pomocą joysticka, tak jak zwykle robiliśmy to na komputerach popularnych w naszym kraju.

Obsługa programu graficznego joystikiem może być dość wygodna, natomiast rysowanie grafiki na zasadzie "piksel po pikselu" jest zupełnie czymś innym niż możliwość swobodnego przesuwania wskaźnika za pomocą powierzchni dotykowej.

> **Opracował:** Marcin Libicki

Epson HX-20

Komputer Epson HX-20 został zaprezentowany w 1981 roku na targach COMDEX, które były organizowane corocznie w latach 1979-2003 w Las Ve-Stanach Ziednocqas w zonych. Były to jedne z największych targów komputerowych na świecie, a w pierwszej edycji udział wzięło ok. 150 wystawców. Sprzęt firmy Epson zwrócił wielką uwagę, bowiem pokazano zupełnie trend miniaturyzacji nowy komputera.

Mam na myśli pomysł na komputer przenośny, który ma wymiary zbliżony do małej walizeczki, posiada solidne możliwości oraz zasilanie bateryjne, co powoduje, że można go używać naprawdę wszędzie. Do tego posiada pełnowymiarową klawiaturę oraz ekran LCD. Te wszystkie cechy posiada właśnie model HX-20.

Komputer posiada wbudowana szybkości 42 linii na minutę. Do zestawu można było dokupić miniaturowy magnetofon kasetowy, który nie ma żadnych przycisków, bo całość transmisji danych jest kontrolowana z poziomu oprogramowania. Ciekawostką jest pojemnośc taśm, która wynosi ok. 50 kilobajtów danych dla kasety 30-minutowej. Wbudowane akumulatory zapewniają żywotność baterii szacowaną przez producenta na około 50 godzin.

Magistrala systemowa (złącze po lewej stronie) pozwala także na dodawanie modułów rozszerzeń, w tym dodatkowej pamięci RAM, gniazd wejścia-wyjścia i innych. Epson produkował też dedykowane plastikowe walizki, aby komputer HX-20 z dodatkowym osprzętem mógł być wygodnie montowany i przenoszony.

Nieco gorzej sytuacja wygląda z oprogramowaniem. Maszyna posiada wbudowane w ROM tylko dwa programy: aplikację służącą do manipulacji pamięcią oraz interpreter języka Microsoft Basic. Z innych zalet warto wymienić możliwość zapisania daty i godziny. Pa-

> trzac na HX-20 zauważymy, że wbudowany ekran jet bardzo mało i mieści tylko 4 linie po 20 znaków, ale na początku lat '80tych był to jeden z największych ekranów LCD jakie mogły znaleźć zastosowanie.

> > **Opracował:** Kamil Stokowski



%5 *6

ZXCVBNM°

Initialize TOR

29

I⁵ O⁶

K

Ρ ~ @

L³ +;

BASIC RTH TTY EMULATOR

7 68

Y U⁴

FGHJ

IBM PC Jr

W tym numerze piszemy o możliwości wykorzystania języka Basic w komputerze PC Jr. Ta nazwa może być myląca, dlatego przyjrzyjmy się tej ciekawej maszynie wprowadzonej na rynek w 1983 roku przez firmę IBM.

PC Jr był próbą wejścia na domowy rynek komputerów IBM. Został zaprezentowany w listopadzie 1983 roku i w wprowadzony do sprzedaży już pod koniec stycznia 1984 roku. Miał mieć wysoki stopień kompatybilności ze standardem IBM PC, który był już wtedy popularnym komputerem biznesowym, oczywiście prócz możliwości stosowania kolorowej grafiki i 3kanałowego dźwięku. Te parametry były lepsze niż zwykła grafika IBM, tak więc PC Jr de facto stał się pierwszym komputerem w historii, który z jednej strony nie zerwał zgodności z IBM PC, a z drugiej - zapewnił kolorową grafikę.

Charakterystyczną cechą było zastosowanie klawiatury bezprzewodowej na podczerwień. Urządzenie standardowo współpracowało z telewizorem przez zewnętrzny modulator zapewniając rozdzielczość 320x240. Standardową pamięć 64 KB lub 128 KB (w zależności od modelu) można było rozszerzać specjalnymi modułami pamięci po 128 KB.

Maksymalnie można było podłączyć cztery takie moduły co rozszerzało RAM do 640 KB, czyli ilości dostępnej w modelu IBM PC XT przeznaczonym na rynek biznesowy. Komputer wykorzystuje systemową do pamięć przechowywania danych wideo, a lokalizacja tego obszaru pamięci może zostać zmieniona. Z tego względu możliwe jest uzyskanie animacji bez uciażliwego efektu migotania i innych funkcji, które były trudne lub niemożliwe do wykonania na ówczesnych klonach peceta.

Mimo to, PC Jr był nisko oceniany przez wielu publicystów oraz firm na rynku komputerowym. Mówiono, że jest to robienie zabawki z "poważnego" komputera, urządzenie nazywano "żałosnym, okaleczonym komputerem", a także wskazywano, że w USA najbardziej zyska na tym firma Apple. Pomimo tych obaw Clappa, kiedy PC Jr stał się powszechnie dostępny na początku 1984 roku, pojawił się spory popyt. Niestety nie był to poziom, o którym firma IBM marzyła. Według informacji producenta, popyt był zmienny i nie rozwijał się "zgodnie z oczekiwaniami", a wielu dystrybutorów miało problemy ze sprzedaniem "Juniora". Dlatego dość wcześnie rozpoczęto oferować rabaty, co jednak nie zmieniło sytuacji, a zapasy komputerów zamiast znikać z magazynów, zaczęły był problemem. Komputery PCjr przestały być produkowane już w 1985 roku i dzisiaj jest to przykład jednej z największych porażek w historii firmy.

Gdzie szukać przyczyn braku popularności? PCjr oparty był procesor Intela i posiadał interfejs BIOS kom-



patybilny z innymi współczesnymi modelami IBM PC, jednak miał też spore różnice w architekturze, które spowodowały brak kompatybilności z dużą ilością oprogramowania. Nie mógł być stosowany jako sprzęt czysto domowy, bo brakowało oprogramowania rozrywkowego, natomiast jako sprzęt biurowy miał problemy z uruchamianiem aplikacji.

Wydaje mi się, że IBM liczył na to, że duża ilość dystrybutorów (ok. 1,5 tys.) oraz popularność standardu biznesowego, będą miało przełożenie na ilość sprzedanych zestawów PC Jr, co z kolei zaowocuje rozwojem rynku oprogramowania domowego, w tym gier. Niestety, w tym czasie mieliśmy do dyspozycji inne komputery 8-bitowe o ugruntowanej pozycji i wyraźnie określonym profilu użytkownika docelowego. Z tego względu PC Jr okazał się sprzętem "dla nikogo". Na popularność w domu pecety musiały poczekać jeszcze kilka lat, a firma IBM była zmuszona do zmiany polityki, aby móc skutecznie konkurować na rozwijającym się rynku maszyn 16-bitowych.

Mariusz Wasilewski

Sega Teradrive

Teradrive to komputer hybrydowy, który łączy możliwości konsoli Sega Mega Drive z w pełni wyposażonym komputerem IBM PC 286. Maszyna została wprowadzona na rynek japoński w 1991 roku, co może być dość zaskakujace, biorac pod uwade ówczesne możliwości pecetów. Procesor 286 to produkt z pierwszej połowy lat '80tych, tak więc firma Sega dość mocno zaryzykowała tworząc nietypowy sprzęt.

Płyta główna Teradrive jest zintegrowana z komponentami peceta, co jest kolejna niespodziewaną cechą. Komputer był porównywany z urządzeniem 386 Amstrad Mega PC, jednak tam możliwości Mega Drive były zapewniane poprzez kartę rozszerzeń w standardzie ISA. Był to po prostu PC z kartą pozwalającą "udawać" konsolę.

Teradrive pozwala również na współpracę pomiędzy jedną i drugą platformą. Odbywa się to w ograniczony sposób, ale jednak istnieje oprogramowanie pozwalające korzystać z tych możliwości. Przykładem może być gra "Puzzle Construction", która zawiera edytor pozwalający na tworzenie własnego zestawu puzzli. Gra oczywiście działa na Mega Drive, ale edytor korzysta z platformy PC.

Z tego, co udało mi się ustalić, wraz z Teradrive w pewnym okresie dołączano zestaw programistyczny SDK, aby zachęcić developerów do tworzenia gier dla japońskiej konsoli. Niestety dzisiaj pakiet jest niedostępny, dlatego mam mały apel do Czytelników: jeśli ktokolwiek ma więcej informacji na ten temat, prześlijcie je proszę do mnie lub na adres redakcji naszego magazynu.

Teradrive zawiera złącze rozszerzeń, które zapewnia taką samą funkcjonalność, jak podobne gniazdo w Mega Drive. Producent planował wydać rozszerzenie pozwalające podłączyć napęd CD, podobnie jak Mega CD (lub Sega CD, bo ten sprzęt był różne nazywany), lecz słaba sprzedaż spowodowała, że Sega skupiła się na rozwoju modelu Saturn.

Warto dodać, że do Teradrive nie można było też w żaden sposób podłączyć rozszerzenia o nazwie 32X, które zostało zaprojektowane jako forma przejściowa przed wydaniem właśnie Segi Saturn w 1994 roku. Dzisiaj oba produkty sa uznawane za komercyjną porażkę, jednak początkowo konsola była bardzo dobrze oceniana przez samych graczy. Z polskiego punktu widzenia model Saturn jest znany głównie z reklam i opowieści, dużo mniej niż przykładowo Amiga CD32. Fakt ten pokazuje, jak mało popularny był u nas japoński sprzęt, bowiem z technicznego punktu widzenia Saturn jest konsolą i dużo większych możliwościach i wiele gier, jak choćby "Sega Rally Championship", nie mogłyby działać na fabrycznej CD32. Ponadto, dla konsoli Commodore wydano tylko ok. 200 gier, a Sega może pochwalić się bazą ok. 1000 tytułów.

Wracając do Teradrive, pod koniec 1993 roku koncepcja sprzętu hybrydowego podchwyciła firma Amstrad, tworząc model Mega PC. Komputer ten był jednak sprzedawany w bardzo wysokiej cenie (ok. 3000 dolarów) , więc trudno się dziwić, że nie uzyskał popularności. Nieco później stworzono model "Plus", który posiadał procesor Cyrix 486DX/33 MHz oraz 4 MB pamięci, ale niewiele zmieniło to pod względem słabej sprzedaży.



Wizyta w gdańskim muzeum RetroKomp

W Gdańsku od ponad dwóch lat działa **Stowarzyszenie** Miłośników Starej Elektroniki i Techniki - RetroKomp. Do organizacji należą między innymi osoby wcześniej prowadzące Szkolne Muzeum Techniki przy Zespole Szkół Łączności w Gdańsku, a także osoby powiązane z organizacią corocznej imprezy RetroKomp, organizowanej cyklicznie od 2012 roku w budynku Gdańskiego Archipelagu Kultury (GAK) w Gdańsku.

Jak głosi jeden z punktów statutu Stowarzyszenia, jego celem jest konserwacja i ochrona zbytków z dziedziny elektroniki i techniki, zwłaszcza produkcji polskiej; zrzeszanie pasjonatów zabytkowego sprzętu elektrotechnicznego oraz wspieranie edukacji osób zainteresowanych w kwestii działania Stowarzyszenia. Organizacja powoli rozrasta się, przyjmowani są nowi członkowie, organizowane są różne imprezy. Stowarzyszenie systematycznie promuje starą elektronikę (w tym także retro-komputery i retro-konsole, które cieszą się największym zainteresowaniem) na różnych interaktywnych wystawach, a także wspiera swoją działalnością różne eventy. Z tegorocznych imprez, które wsparło Stowarzyszenie prezentując i udostępniając zwiedzającym swoje sprzęty, można wymienić: Szczecinek Retro Games, Szlamfest 2: Zemsta Szlamu, Rumia Comic Con, czy Europejską Noc Muzeów. W lipcu tego roku zostało także otwarte, i odremontowane, muzeum w Gdańsku i to o nim napiszemy w niniejszym artykule.

Jako, że w północnej części Polski, nie ma za wiele tego typu miejsc (mamy co prawda koszalińskie Muzeum Elektroniki Komputerowej ELKON, czy toruńskie Muzeum Informatyki na Uniwersytecie Mikołaja Kopernika) i mogłoby się wydawać, że na południu Polski występuje więcej tego typu obiektów (wśród nich znajdziemy chociażby warszawskie Muzeum Techniki i Przemysłu, Muzeum Historii Komputerów i Informatyki w Katowicach, czy Muzeum Konsol Gier Video w Karpaczu) to właśnie muzeum RetroKomp jest raczej nietypowe i wyróżnia się na tle pozostałych tego typu miejsc. Znajdziemy tu bowiem oprócz sali typowo komputerowej (w której poza komputerami znajdują się również wszelkiej maści konsole) także salę z elektroniką użytkową (telewizory, radia, odtwarzacze, gramofony itd.), oraz salę kinową z "obwoźnym" projektorem i fotelami z byłego kina Neptun (dawniej Leningrad) w Gdańsku. Nadmieńmy, że kino Neptun mieszczące się kiedyś przy ulicy Długiej w Gdańsku zakończyło swoją działalność jakiś czas temu (ostatni, pożegnalny seans odbył się w 2015 roku) i obecnie w jego miejscu został wybudowany hotel. W budynku muzeum jest także magazynek, gdzie znajdują się zapasowe sprzęty, nadmiarowe telewizory i części zamienne.

Warto nadmienić, że muzeum nie jest jedynie wystawą z gablotami, obwieszoną zakazami typu:"Prosimy nie dotykać eksponatów", tylko ma formę interaktywną. Tutaj każdy odwiedzający, jeśli tylko chce, może sprzęt dotknąć, uruchomić, pograć na nim, a nawet napisać jakiś program komputerowy. Przychodzą tutaj zarówno dzieci j





młodzież, poznając historię elektroniki i informatyki, oraz starsi odwiedzający, którzy mieli kiedyś styczność z tego rodzaju sprzętem i z łezką w oku przypominają sobie stare czasy.

.Jak wspomnieliśmy na początku artykułu, jedną z muzealnych atrakcji jest niewielka sala kinowa. W jej centrum, przy ścianie stoi działający i odrestaurowany projektor KH17 produkcji radzieckiej z roku 1970. Projektor był urządzeniem używanym swego czasu w kinie objazdowym. W latach PRL, w małych miasteczkach i wsiach pojawiały się takie mobilne kina, gdzie nie było dostępu do dużvch stacjonarnych kin. Na miejsce przyjeżdżał pojazd z niezbędnym oprzyrządowaniem, a kinooperator zapewniał ludziom rozrywkę, nierzadko wyświetlając filmy propagandowe. Ten konkretny projektor, zmagazynowany był w kinie Neptun, a członkom Stowarzyszenia udało się go pozyskać na cele muzealne. W sali kinowej muzeum znajduje się także stół do przewijania taśm filmowych (ich długości mogą sięgać nawet kilku kilometrów). W muzeum co jakiś czas prezentowane są pokazy trailerów filmowych z różnych lat, zapisanych na taśmach 35mm. Na tych taśmach zapisana jest także ścieżka dźwiękowa, niestety projektor KH17 go nie odtwarza, mimo że posiada służące do tego oprzyrządowanie. Powodem tego, jest brak dodatkowego modułu wzmacniającego dźwięk. Być może w przyszłości będzie on domontowany do projektora. Warto też zwrócić uwagę na pięknie odrestaurowane cztery fotele, które tak jak projektor, zostały pozyskane z byłego kina Neptun.

Kierując swoje kroki dalej, przechodzimy do sali RTV. Tutaj na podłodze i na

stołach stoja stare kineskopowe telewizory, radioodmagnetofony, biorniki, gramofony itd. Typowa elektronika użytkowa z lat 70. i 80. zeszłego wieku. W tym pomieszczeniu nic nie jest jednak włączone - jest to bardziej wystawa urządzeń z dawnych lat. Chociaż jak zapewniają członkowie Stowiększość warzyszenia z tych eksponatów jest nadal sprawna. Jedną z ciekawostek jest magnetowid szpulowy MTV-10, którego produkcja rozpoczęła się na początku lat 70. a więc jeszcze przed erą VHS. Producentem urządzenia były Zakłady Radiowe im. Marcina Kasprzaka w Warszawie. Magnetowid potrafił zapisać i odczytać czarno-biały obraz wideo, oraz dźwięk. Swoją konstrukcją bazuje na urządzeniu LDL 1001 marki Philips.

Ostatnią, przy tym największą salą, jak i prawdopodobnie cieszącą się największym zainteresowaniem, jest sala komputerowa. W niej znajdziemy zarówno wystawę starych konsol i komputerów (cieszy oko cały zestaw konsol produkowanych przez firmę Atari, gdzie znajdziemy modele 2600, 5200, 7800, Jaguara, jak i dwa modele przenośnej konsoli Atari Lynx), oraz gablota z



Sala kinowa.

akimi okazami jak terminal AN-2000, Bosman-8 z terminalem ANG-3001, WANG, Commodore PET 2001, Commodore Plus/4, czy Unipolbrit 2086. Sam terminal AN-2000 jest bardzo ciekawym urządzeniem i zasługuje na więcej słów. Jest to prototypowy terminal ekranowy polskiej produkcji (obudowę jak i elektronikę zaprojektowano na Politechnice Gdańskiej) z własnym procesorem. Terminal łączy się przewodowo, z zewnętrznym komputerem typu mainframe (uniwersalna, wielozadaniowa jednostka, do której ma dostęp kilku-kilkunastu użytkown-



Sala komputerowa.



Sala RTV.

ików). Terminal ten pozwalał na wprowadzanie danych z klawiatury do systemu i wyświetlanie na ekranie podłączonego do terminala monitora informacji zwrotnej.

Oprócz modeli za szybą, na stołach ustawionych wokół sali są porozkładane, podłączone i gotowe do uruchomienia konsole i komputery, podpięte pod stare telewizory i monitory CRT. Wśród nich znajdziemy dobrze znane 8-bitowce jak ZX Spectrum, Atari XL/XE, Commodore 64, mniej znany Elwro 800, jaki i 16-bitowce takie jak Amiga 500 (jest także model CDTV), Atari 520ST, a także konsole: Sega Mega Drive, Nintendo Entertainment System i wiele innych.

Na wyposażeniu Stowarzyszenia jak i muzeum jest także auto dostawcze marki Daewoo Lublin z 1998 roku, które służy jako objazdowy, interaktywny retro-kącik (auto można było zobaczyć m.in. na Pixel Heaven 2017). Auto pozyskano z Agencji Mienia Wojskowego i przerobiono tak, aby można było w środku podłączyć kilka dowolnych komputerów do zamontowanych na ścianie na stałe telewizorów LCD. Na wyposażeniu tego pojazdu, jest agregat prądotwórczy, który zapewnia zasilanie dla retro komputerów.

Dzięki takiej zróżnicowanej wystawie, mobilności i interaktywności, muzeum cieszy się sporym zainteresowaniem. Odwiedzający, przychodząc tu tylko na chwilę z ciekawości, zostają na dłużej, wciągając się z rozmowy z obsługą, ogladaja, chca zobaczyć jak najwiecej. To też było celem Stowarzyszenia, aby nie poprzestawać na kilku jednodniowych pokazach w roku, chowając na resztę czasu sprzęt w szafach i magazynach, tylko udostępnić w miarę możliwości ciągłą wystawę, aby każdy w dogodnym dla siebie terminie, mógł muzeum odwiedzić i poznać trochę historii elektroniki. Póki co muzeum otwarte jest jedynie w weekendy, w godzinach: sobota 16-20, niedziela 14-18. W razie potrzeby zorganizowania wycieczek do muzeum w inne dni, lub współorganizacji imprez wraz ze Stowarzyszeniem, można kontaktować się z jego członkami (kontakt na końcu tekstu). Osoby zainteresowane wsparciem muzeum czy to w postaci finansowej, lub chętne na przekazanie starego sprzetu elektronicznego, także zapraszane są do kontaktu. Można także zapisać się do Stowarzyszenia, jeśli ktoś byłby chętny. Celem muzeum jest

dalsze pozyskiwanie ciekawych eksponatów, prezentowanie starej elektroniki na kolejnych imprezach i rozbudowa wystaw w samym muzeum. Zapraszamy do odwiedzania!

Michał "stRing" Radecki-Mikulicz

MUZEM RETROKOMP

http://muzeum.retrokomp.org/

https://www.facebook.com/StowarzyszenieRetroKomp/

ul. Kotwiczników 10AB (dojazd od ulicy Lastadia) 80-881 Gdańsk



Logo w szkole

Język logo jest dzisiaj praktycznie zapomniany, a został specjalnie stworzony po to, aby edukować młode osoby zainteresowane informatyką. Został zaprojektowany w latach '60-tych i składa się z gotowych prostych funkcji, które służą do definiowania własnych procedur przez użytkownika. Język korzysta z tak zwanej grafiki żółwia, co ma powodować uproszczenie obsługi oraz łatwość obserwacji działania poszczególnych poleceń.

Wiele lat temu mojej szkole podstawowej język Logo był wykorzystywany bardzo często na zajęciach koła informatycznego, a później już regularnych lekcji podstaw informatyki. Można było zobaczyć w akcji komputery Elwro Junior 800 i choć nie był to dla mnie pierwszy kontakt z elektroniczną maszyną, programowanie w Logo na chwilę stało się ważne. Jednak nigdy nie zastanawiałem się nad historią tego języka programowania, a jest o czym mówić.

Logo jest oparte jest na Lisp, jednak posiada zupełnie inną składnię. Lisp to język programowania, który został zaprojektowany w 1958 roku. Jest to drugi najstarszy język programowania wysokiego poziomu, który nadal pozostaje w użyciu - starszy jest tylko Fortran. Podobnie jak ten ostatni, Lisp ulegał na przestrzeni czasu licznym zmianom, powstało również wiele jego dialektów.

Lisp w gruncie rzeczy powstał jako elastyczny sposób zapisywania działań

matematycznych w programach komputerowych. Jednak szybko został najchętniej wybieranym językiem do badania i rozwoju projektów takich jak na przykład sztuczna inteligencja. Wywodzi się z niego wiele technik programistycznych, które używamy także dzisiaj, na przykład koncepcja programowania obiektowego.

Nazwa Lisp pochodzi od słów "List Processing". Podstawową strukturą danych jest lista, natomiast kod źródłowy programów w Lispie składa się z wielu list. W wyniku tego programy mogą manipulować kodem źródłowym jak zwykłą strukturą danych. Umożliwia to pisanie tak zwanych makr, pozwalających programiście tworzyć nową składnię lub nawet małe, własne języki, które są zagnieżdżane w Lispie.



Przykłady działania instrukcji języka Logo, w wersji angielskiej oraz polskiej.

Kod tworzony jako struktura danych sprawia, że Lisp ma charakterystyczną składnię. Cały kod źródłowy ma postać list otoczonych nawiasami. Wywołanie funkcji lub makra ma postać listy, której pierwszym elementem jest nazwa funkcji, a następnymi elementami – jej argumenty. Na przykład funkcję o nazwie F z argumentami A, B i C wywołuje się za pomocą kodu "(f a b c)".

Wracając do Logo, jest to w pewien sposób ewolucja powstała od czasów Lispa. Wspomniana grafika żółwia wywodzi się od czasów początków tego języka, gdy służył on do sterowania robotem zwanym, ze względu na swój wygląd, właśnie żółwiem (ang. turtle). Robot ten po wpisaniu komendy takiej jak FORWARD 50 przemieszczał się po podłodze o pięćdziesiąt kroków lub obracał się w prawo o kąt prosty po komendzie RIGHT 90. Nasz żółw wyposażony był także w specjalne pióro, za pomocą którego mógł znaczyć trasę swojej wędrówki lub przemieszczać się bez "rysowania" swojej drogi.

Wraz z upływem czasu, gdy komputery zaczęły używać monitorów i grafiki, żółw Logo został przeniesiony na ekran monitora. Żółw na ekranie jest znacznie tańszy w eksploatacji, szybszy oraz posiada znacznie więcej możliwości niż jego fizyczny odpowiednik. Mimo to, jest to technika nadal interesująca, chociażby ze względu na możliwość pomiaru właściwości otoczenia, gdy żółw jest wykorzystywany w terenie.

Jak widać, nie jest to więc jedynie zabawa edukacyjna, choć w moim wypadku szkoła stwarzała takie pozory. Nauczyciel omawiał zarówno angielską, jak i polską wersję Logo, gdzie można było stosować rodzime odpowiedniki instrukcji, na przykład PRAWO zamiast RIGHT. Dzięki luźnej atmosferze zajęć, wiele osób chętnie uczyło się rysowania po ekranie za pomocą żółwia, co stanowiło doskonałe wprowadzenie do programowania. Rzecz jasna, dzisiaj powiemy, że są lepsze sposoby, ale w latach '80-tvch trudno było sobie wyobrazić bardziej uniwersalny sposób wyjaśnienia na czym polega programowanie dzieciom. Tym bardziej, że produkty firmy Elwro były zgodne ze standardem ZX Spectrum, a dodatkowo można było na nich uruchomić system CP/M.

Pamiętam, że wielkie wrażenie robił na mnie komputer, który pozwalał uruchamiać gry, a w tym samym czasie pracować w sieci ze współdzieloną drukarką i uruchamiać "profesjonalne" oprogramowanie przypominające system MS-DOS na pececie. Narysowane obrazy w Logo można było wydrukować na drukarce znajdującej się w zupełnie innym pomieszczeniu, bez potrzeby zapisywania programu na taśmie czy dyskietce. Pewnym problemem były tylko częste awarie, z którymi nauczyciel radził sobie samodzielnie i do tego na tych samych zajęciach koła informatycznego. Po prostu siadał, rozkręcał sprzęt i próbował go naprawić. Przy okazji wiele osób pierwszy raz w życiu mogło zobaczyć wnętrze prawdziwego komputera oraz przekonać się, że używanie tej nowinki technicznej wiąże się także z kłopotami i trzeba umieć sobie z nimi dać radę. Myślę, że nie powinniśmy przeceniać takiej "nauki na żywym organizmie", ale nie należy także jej nie doceniać, co dzisiaj zdarza się bardzo często.

Logo doczekało się licznych implementacji, na przykład Logo Komeniusz, który powstał dla systemu Windows. Pozwala on na tworzenie bardziej rozbudowanych programów, jak również definiowanie zmiennych, pojęć matematycznych, figur geometrycznych, funkcji itp. Program posiada ekran graficzny, gdzie powstają grafiki rysowane przez żółwia, oraz obszar tekstowy, gdzie wpisujemy polecenia. Do tego miejsca wygląda to podobnie jak w klasycznym logo. Komeniusz ma również tryb pracy interaktywnej, w którym można wydawać polecenia i obserwować skutki ich działania bezpośrednio, a także tryb definiowania obiektów pozwalający tworzenie procedur w osobnym edytorze.

Program Komeniusz dostępny jest także w wersji on-line na poniższej stronie:

http://logokom.wiktor.be/



Procedury dźwiękowe

Sterowanie dźwiękiem ZX Spectrum jest bardzo proste i wymaga właściwie tylko poznania instrukcji BEEP. W tym artykule chciałbym pokazać, jak można to wykorzystać do zaprogramowania całkiem ciekawych efektów, jedynie korzystając z Basica.

Jak wiadomo, dźwięk jest odtwarzany za pomocą słowa kluczowego BEEP, a jego ogólny sposób użycia jest następujący:

BEEP czas trwania, wysokosc

Jako 'czas trwania" i "wysokość" wpisujemy liczby. Pierwszy argument podawany jest w sekundach, a drugi - w półtonach powyżej środkowego dźwięku C (liczby dodatnie) lub dla nut poniżej środkowego C (przy użyciu liczb ujemnych).

Aby uzyskać wyższe lub niższe dźwięki, musimy dodać lub odjąć wartość 12 dla każdej oktawy, którą chcemy uzyskać - w górę lub w dół. Prosty przykład widać w ramce obok.

Nie są to przypadkowe nuty, lecz fragment marszu pogrzebowego z pierwszej symfonii Gustava Mahlera. Zaskakujące, prawda? Tak właśnie programowano dźwięk w latach '80-tych i chciałbym nawiązać do tego czasu. Załóżmy, że melodia jest zapisana w tonacji C-mol, co na pięciolinii wygląda tak:



Przepisując to do programu komputerowego powinnismy zastosować poniższy schemat :



Jeśli chcesz zmienić tonację, najlepiej jest ustawić dodatkową zmienną i dodawać ją do każdego elementu w programie. Przykładowo może to wyglądać następująco:

```
20 BEEP 1,tonacja+0:
BEEP 1,tonacja+2:
BEEP s,tonacja+3:
BEEP.5,tonacja+2:
BEEP 1,tonacja+0
```

Przed uruchomieniem programu należy podać odpowiednią wartość zmiennej "tonacja":

ø	- dla C-moll,
2	- dla D-moll,
12	- dla C-moll
	(oktawa w górę)

i tak dalej. W ten sposób dopasujemy dźwięki ze Spectrum do innego instrumentu, trzeba tylko używać wartości ułamkowych.

Oprócz wysokości dźwięku trzeba sprawdzić czas trwania wszystkich nut. Można przyjąć wartość jednej sekundy dla ćwierćnuty i oprzeć na tym cały odgrywany utwór. Wtedy ósemce przyporządkujemy pół sekundy i tak dalej.

Bardziej wygodne jest - tak jak poprzednio - wprowadzenie dodatkowej zmiennej, co pozwoli na szybkie zmiany tempa melodii. W tym celu naszą linię 20 należy zmodyfikować tak jak poniżej:

```
20 BEEP czas,tonacja+0:
BEEP czas,tonacja+2:
BEEP czas/2,tona-
cja+3:
BEEP czas/2,tona-
cja+2:
BEEP czas,tonacja+0
```

Pamiętajmy, że w komputerze jest tylko jeden głośnik, dlatego można zagrać tylko jedną nutę jednocześnie. Aby zorientować się w wysokości poszczególnych dźwięków można wpisać pętlę podobną do poniższej:

```
FOR n=0 TO 1000
BEEP .5,n
NEXT n
```



STREF# 2X SPECTRUM

10 BEEP 1,0: BEEP 1,2: BEEP .5,3: BEEP.5,2: BEEP 1,0
20 BEEP 1,0: BEEP 1,2: BEEP .5,3: BEEP.5,2: BEEP 1,0
30 BEEP 1,3: BEEP 1,5: BEEP 2,7
40 BEEP 1,3: BEEP 1,5: BEEP 2,7
50 BEEP .75,7: BEEP .25,8: BEEP .5,7: BEEP .5,5:BEEP .5,3: BEEP.5,2: BEEP 1,0
60 BEEP .75,7: BEEP .25,8: BEEP .5,7: BEEP .5,5: BEEP .5,3: BEEP .5,2: BEEP 1,0
70 BEEP 1,0: BEEP 1,-5: BEEP 2,0
80 BEEP 1,0: BEEP 1,-5: BEEP 2,0

Spowoduje to odegranie wszystkich możliwych dźwięków, a następnie program zatrzyma się na skutek wywołania błędu. W tej sytuacji wyświetlenie zawartości zmiennej N daje możliwość sprawdzenia, ile dźwięków zostało odtworzonych prawidłowo.

Tego samego sposobu omżna spróbować dla niskich dźwięków, ale wiele z nich będzie brzmiało jak kliknięcia. W związku z tym, tylko środkowy zakres nut będzie naprawdę przydatny. Niskie dźwięki brzmią zbyt mocno, natomiast wysokie dźwięki są zbyt piskliwe i mają tendencję do lekkiego falowania.

Wpisz tę linię programu:

tosowując wszystkie dźwięki, aby brzmiały bardziej przyjemnie dla ucha. Może to zrobić, poruszając palcami w górę lub w dół struny, co oczywiście nie jest możliwe na fortepianie. Naturalną skalę, którą gra skrzypek, można uzyskać za pomocą następującego programu: równa połowie półtonu, będzie miała wartość 0,5.

Możemy też sprawić, aby naciśnięcia klawiszy powodowały odgrywanie dźwięku, zamiast standardowych kliknięć. Wystarczy wpisać poniższą instrukcję:

POKE 23609,255

Druga liczba określa długość dźwięku. Można tu wpisywać różne wartości od 0 do 255.

Myślę, że najlepszą metodą na lepsze poznanie możliwości komputera są samodzielne eksperymenty. Nawet jeśli napiszemy program tylko w Basicu, może on prezentować interesujące algorytmy. W artykule starałem się zebrać informacje o tym, jak zacząć zabawę z dźwiękiem, który 30 lat temu robił na mnie wielkie wrażenie.

```
20 BEEP .5,0: BEEP .5,2,039: BEEP .5,3,86:
BEEP .5,4,98: BEEP .5,7,02: BEEP .5,8,84:
BEEP .5,10,88: BEEP .5,12: STOP
```

Możemy tutaj nie zauważyć żadnej różnicy, bowiem wartości różnią się bardzo niewiele. Można powiedzieć, że jest to przykład dla perfekcjonistów, ale warto wiedzieć, czym różnią się obie

10 BEEP .5,0: BEEP .5,2: BEEP .5,4: BEEP .5,5: BEEP .5,7: BEEP .5,9: BEEP .5,11: BEEP .5,12: STOP

Odgrywa ona skalę C-dur, która wykorzystuje wszystkie białe klawisze na fortepianie, od środkowego C do następnego C w górę. Sposób, w jaki ta skala jest dostrajana, jest dokładnie taki sam jak na fortepianie i jest nazywany równomiernie temperowanym, ponieważ interwał półtonu jest taki sam na całej skali.

Jednak przykładowo skrzypek będzie grał przy tej skali nieco inaczej, dos-

skale i jak je zastosować w programie w Basicu.

Niektóre rodzaje muzyki - w szczególności muzyka indyjska - używają odstępów mniejszych niż półton.

Możemy je zaprogramować za pomocą instrukcji BEEP bez żadnych problemów, lecz musimy użyć innych wartości interwału, na przykład ćwierćton, czyli odległość między dźwiękami Dzisiaj możemy być zdziwieni, jak programiści sprzed lat radzili sobie z programowaniem prostego głośnika wydającego pojedyncze piski.

Wiele informacji na temat programowania w Basicu na ZX Spectrum można znaleźć także na poniższej stronie internetowej:

http://www.worldofspectrum.org/ZXBasicManual/.

Natomiast krótkie omówienie instrukcji, ale za to w języku polskim, znajduje się tutaj:

https://fizyka.umk.pl/~jacek/zx/doc/basic.htm

> Opracował: Kamil Stokowski

리며



ZX-81 i gry 16 KB

Komputer ZX-81 był bardzo popularny w USA, gdzie sprzedano ok. 300 tys. egzemplarzy. Bardzo duży nacisk na komputeryzację był również w Wielkiej Brytanii i tam ZX-81 odgrywał nie mniejszą rolę. Dlatego duża część oprogramowania pochodzi właśnie z Wysp Brytyjskich. Chciałbym przedstawić dwa produktv tamtejszej firmv Softsync, która sprzedawała gry Meteorites i Red Alert. Oba tytułu są bardzo grywalne, mają szybką akcję i wymagają rozszerzenia pamięci RAM o wielkości 16 kilobajtów.

METEORITES

Pierwsza gra jest wzorowana na typowej grze automatowej. Musisz bronić swojego statku kosmicznego przed skałami kosmicznymi. Sterujemy nim za pomocą klawiatury. Aby dobrze wycelować działo statku, naciśnij przycisk "6", aby obrócić statek w kierunku przeciwnym do ruchu wskazówek zegara, a przycisk "7", aby obrócić zgodnie z ruchem wskazówek zegara. Naciśnięcie klawisza "0" uruchamia torklawisza "9" Naciśnięcie pedy. powoduje odpalenie silników i porusza statkiem w dowolnym kierunku.

Ze względu na niską rozdzielczość grafiki, nie można było przedstawić dokładnym kształcie statku 0 graficznym. Zamiast tego gra używa liczb od "1" do "8", aby przedstawić statek i jego ustawienie. I tak, że znak "1" oznacza, że statek ma orientację "na północ", "2" oznacza statek skierowany na północny wschód, "3" wschód i tak dalej. Może to brzmieć nieco dziwnie, ale w trakcie rozgrywki można się dość szybko przyzwyczaić do tego nietypowego sposobu prezentacji graficznej.

Meteoryty są widoczne jako kształty graficzne i dzielą się na pięć znaków "0" po trafieniu przez torpedy. Mniejsze kawałki są następnie odrzucane przez nasze kolejne trafienia. Obiekty mogą opuścić jedną stronę ekranu i pojawić się z drugiej, ale jest to typowa cecha tego typu gier.

Efekty graficzne są rzadkie i nie ma żadnych wymyślnych eksplozji, to cała gra jest swojego rodzaj u hołdem dla ZX-81, bowiem jest napisana w Asemblerze, a kod źródłowy zajmuje tylko 3 kilobajty w pamięci.

Na przykład stopień trudności jest kontrolowany przez serię instrukcji POKE zmieniających wartości w określonych adresach pamięci. Możesz na przykład zwiększyć liczbę punktów, które przyz-





STREF® 2X SPECTRUM

6-DOWN 7-UP 9-FIRE 0-BOMB FUEL 982 SCORE 0000 HIGH 0000



Gra wygląda na bardzo prostą, ale działa bardzo szybko i dynamicznie.

nawane są dla statków, zmieniać liczbę statków lub zmienić skok strzału torpedy. Szczegóły pozostawiam graczom zainteresowanym technicznymi aspektami uruchamianych tytułów.

RED ALERT

Ta gra przypomina popularną grę zręcznościową Defender. Losowy krajobraz gór i dolin przewija się poziomo u dołu ekranu, nadając ruch Twojemu statkowi. Można go przesuwać w górę lub w dół za pomocą klawiszy kursora (klawisze "7" i "6"). Kontrola jest za-



skakująco szybka i wymaga trochę praktyki, aby nie uderzać w górną krawędź ekranu - zwłaszcza, że każdy nieszczęśliwy wypadek od razu rozsadza statek na kawałki.

Obce statki ciągle przesuwają się po ekranie. Twoi wrogowie próbują Cię zniszczyć na trzy sposoby: wystrzeliwując pociski, ładunki kamikadze i próbując skierować Cię w górną lub dolną część ekranu. Oczywiście możesz zrobić coś więcej niż tylko uciekać. Można próbować uników, ale jest to dość denerwujące, bo szybkość akcji powoduje,

> że bardzo łatwo można polecieć "zbyt wysoko", nawet jeśli ominiesz wysoką górę i statki obcych.

Dodatkowymi elementami są obce bazy na ziemi, które pozwalają uzyskać dodatkowe punkty. Odważni i zręczni piloci mogą przemierzać powierzchnię, wysadzając obce bazy pociskami. Jest to ryzykowny manewr, ponieważ Twój statek przy najmniejszym błędzie rozlatuje się na kawałki.

Niektóre bazy są w dolinach, co zmusza do używania innego rodzaju uzbrojenia - bomb. Naciśnięcie klawisza "9" powoduje natychmiastowe wypuszczenie jednej lub kilku bomb, niszcząc wszystko wokół. Znowu wymaga to praktyki, aby nauczyć się poprawnie "naprowadzać" cele.

Na powierzchni można spotkać także obce statki, które należy zniszczyć zanim wystartują. Inaczej piloci kamikadze kierują się wprost na Twój statek kosmiczny i bardzo trudno uniknąć zderzenia.

Niezależnie od prostoty grafiki używanej w obu grach warto zauważyć, jak szybko działają na komputerze o tak ograniczonej wydajności. Na ekranie mamy szybką akcję, precyzyjne sterowanie oraz wiele obiektów animowanych jednocześnie. Byłoby to niemożliwe, gdyby tytuły te zostały napisane w Basicu i bardzo dobrze wykorzystują możliwości ZX-81. Różnice prędkości są widoczne tym bardziej, im wolniejszy sprzęt mamy do dyspozycji, ale podobne przyspieszenie można uzyskać na każdej platformie. Język maszynowy ma w tym wypadku bardzo szerokie zastosowanie i uważam, że powinniśmy z niego korzystać jak najczęściej.

Czy warto grać na tak prostym komputerze jak ZX-81? Oczywiście nie możemy się spodziewać fajerwerków graficznych, ale zawsze można docenić kunszt programistyczny, szybkość pracy czy nietypowe pomysły w przedstawionych produkcjach. Bardzo ciekawe jest także uruchomienie tytułów, które wymagają tylko 1 kilobajta pamięci. Przykładowy opis takiej kolekcji można znaleźć tutaj:

http://www.zx81stuff.org.uk/ zx81/tape/1KGames(Sinclair)



Pierwszy program w języku maszynowym

Napisanie pierwszego programy w danym języku zawsze wiąże się z zadowoleniem, choć szybko przychodzi refleksja, ile jeszcze nie wiemy. Chciałbym pokazać, jak można napisać naprawdę najprostszy program w Basicu, a później przełożyć do na język maszynowy.

Większość początkujących programistów pisze programy typu "Hello World" i zwykle wygląda to podobnie jak poniżej:

10 PRINT "Hello World" 20 GOTO 10

Oczywiście tekst może się różnić, ale rezultat działania jest ten sam - wyświetlamy na ekranie prosty tekst. Jest to podstawowa umiejętność i w przypadku Basica nie ma nic prostszego. Jednak w jaki sposób dokonać konwersji powyższego programu na kod maszynowy?

Aby wypisać ciąg znaków na ekranie, musimy wywołać dwie procedury jedną, aby otworzyć górną część ekranu jako kanał 2, a następnie drugą - wyświetlającą tekst. Kanał możemy otworzyć za pomocą procedury ROM znajdującej się w adresie 5633. Numer kanału ładujemy do akumulatora, a następnie za pomocą adresu 8252 wyświetlamy ciąg tekstowy określając oddzielnie początek (adres w pamięci) oraz jego długość.

Po otwarciu kanału 2 wyjście jest kierowane na górny ekranu, dopóki nie wywołamy adresu 5633 z inną wartością, aby wysłać dane wyjściowe w inne miejsce. Inne interesujące kanały to 1 dla dolnego ekranu, np. PRINT #1 w Basicu. Możemy go użyć, aby wyświetlić tekst na dwóch dolnych liniach.

Uruchomienie programu zamieszczonego w ramce poniżej powoduje wypełnienie ekranu tekstem. Następnie pojawi sie standardowe pytanie "Scroll?". Należy jednak zauważyć, że nowy tekst nie pojawia się w kolejnych liniach, lecz zaraz obok poprzedniego napisu, co nie jest takim samym rezultatem jak program w Basicu.

Aby to zmienić trzeba dodać kod ASCII o wartości 13 na końcu tekstu, co spowoduje, że końcówka programu będzie wyglądała tak:

string defb 'Prgramowanie jest proste' defb 13 eostr equ \$

Istnieje wiele kodów sterujących ASCII, które zmieniają bieżącą pozycję wyświetlania tekstu, kolory itp. Oto kilka przykładów:

16 - ustawia kolor tekstu na wartość następującego bajtu.

17 - ustawia kolor tła na wartość następującego bajtu.

22 - ustawia współrzędne X i Y tekstu na wartości podane w dwóch kolejnych bajtach.

Z powyższych kodów można korzystać analogicznie do ASCII numer 13, tak więc w łatwy sposób uzyskujemy nowe funkcje.

```
ld a,2 ; gorny ekran
call 5633 ; otwieramy kanal
loop ld de,string ; adres ciagu tekstowego
ld bc,eostr-string ; dlugosc ciagu
call 8252 ; wyswietlany tekst
jp loop ; powtarzamy wywsietlanie
string defb 'Programowanie jest proste'
eostr equ $
```

Uruchamiamy system FutureOS

FutureOS to bardzo ciekawym systemem operacyjnym dla komputerów Amstrad CPC 6128 i CPC 464. Jest to bardzo nowatorski projekt napisany w całości w Asemblerze Z80. System jest kompatybilny ze starymi formatami dyskowymi oraz formatami plików, włączając w to pliki binarne. Zawiera on różne narzędzia do obsługi dysków i plików, a także ma wiele funkcji niedostępnych na innych komputerach 8-bitowych.

System może być zapisany na dyskietce lub w formie układów pamięci EPOM. Na dyskietce mamy cztery pliki o objętości 17 KB każdy. Symbolizują one jeden z czterech ROM-ów określanych jako A, B, C i D. Nazwa pliku zawiera numer ROM. Na przykład: "T-OG-E-A.ROM" oznacza ROM A.

Na początek musisz zainstalować cztery pliki ROM, co można zrobić za pomocą programu "--RUN-ME.BAS". Pliki muszą być zainstalowane z właściwym numerem, inaczej FutureOS nie będzie działał.

Jeśli system mamy zapisany w układach EPSON, mogą to być 4 kości 16-kilobajtowe lub 2 kości 32-kilobajtowe. Zależy od Twojej karty EPROM. Każda pamięć EPROM o rozmiarze 16 KB wymaga osobnego numeru pamięci ROM. Pamięć musi być zainstalowana w odpowiednich gniazdach EPSON, a następnie należy przypisać prawidłowy wybór zawartości ROM. Numery pamięci można wybierać za pomocą zworek lub korzystania z odpowiedniego gniazda.

Po instalacji FutureOS musisz wykonać reset. Standardowy tekst po włączeniu komputera powinien wyglądać jak zwykle. System dodaje jednak cztery linie, na przykład na CPC 6128 powinno wyglądać to następująco:

Future OS (oG) Turbo Desk ROM aktive! Future OS (oG) Utility-ROM aktive! Future OS oG/+ Floppy ROM aktive! Future Operating System oG c1989-2007

Jeśli brakuje jednego z powyższych wierszy, ozna cza to, że musiał wystąpić błąd podczas instalacji. Przed ponownym rozpoczęciem instalacji należy wyłączyć wszystkie inne rozszerzenia, wykonać reset i ponownie sprawdzić połączenia.

Po udanej instalacji możemy od razu pracować z FutureOS. Wystarczy wpisać polecenie |OS lub |FDESK, aby uruchomić system. Użycie instrukcji |FDESK powoduje, że FutureOS może być wywołany z poziomu Basica jako podprogram. W takim przypadku możesz powrócić do interpretera Basica z niezmienioną zawartością 48 KB RAM, czyli wpisany listing nie zostanie stracony. Za pomocą polecenia |FVER (lub |FHELP) możesz wyświetlić aktu-



alną wersję swojej instalacji FutureOS. Otrzymasz datę ostatniej modyfikacji i trochę więcej informacji dotyczących systemu.

Po wpisaniu |OS lub |FDESK w Basicu, system jest aktywowany i zobaczysz pulpit. W górnej części ekranu znajdują się cztery linie ikon. Każda linia zawiera az siedem pozycji. Dolna połowa ekranu służy do wyświetlania katalogów dysków. W lewym górnym rogu ekranu zobaczysz wskaźnik, który można przesuwać za pomocą joysticka, klawiszy kursora, a także myszy lub trackballa. Jako przycisk myszy działa przycisk Fire w joysticku. Klawisz ESC powoduje wyjście z każdej uruchomionej funkcji. Wskaźnik umożliwia również oznaczanie plików, trzeba tylko umieścić strzałkę nad plikiem i naciśnąć Fire.

Ikony napędów posiadają duże znakami wewnątrz. Symbole zaczynają się od litery A, a kończą na M. Każdy znak określa nośnik służący do przechowywania danych. Litera A oznacza wewnętrzny 3-calowy napęd dyskietek, B symbolizuje napęd dyskowy podłączony do portu Disk Drive 2. Ikony C i D oznaczają dwa dodatkowe dyski. Litery E i F symbolizują dwie stacje 5,25- i 3,5-calowe działające przy użyciu zewnętrznego kontrolera dysków Vortex.

Cztery partycje dysku twardego są symbolizowane ikonami I, J, K i L. Ikona M oznacza RAM-dysk. Ikona przyciemniona oznacza, że wybór nie jest możliwy, bo urządzenie aktualnie jest nieaktywne. Wybór napędu odbywa się oczywiście poprzez "kliknięcie" na ikonę, wtedy grafika staje się podświetlona. Pulpit systemu FutureOS wygląda na trudny w obsłudze i mnie osobiście przypomina nieco symbole stosowane w Calamusie na Atari.

Nie jest to bardzo czytelny ekran, tak więc poruszanie się po nim wymaga przyzwyczajenia. W następnym numerze naszego magazynu postaram się pokazać, jak używać podstawowych komend systemowych.

> Opracował: Marcin Libicki



Wybrane cechy systemu FutureOS:

- odczytywanie i wyświetlanie zawartości wielu katalogów (do 8 dysków i 4 partycji na dysku twardym).
- wyświetlanie atrybutów plików, brak ograniczeń rozmiaru pojedynczego pliku,
- wyświetlanie nagłówków plików oraz ikon,
- wczytywanie i zapisywanie plików o maksymalnej objętości nawet 4 MB,
- możliwość zmiany nazwy plików, wraz z rozszerzeniem,
- usuwanie wielu plików na raz, nawet zapisanych na różnych dyskach,
- formatowanie i kopiowanie dysków,
- ustawianie czasu i daty.

Pełna dokumentacja systemu jest dostępna formacie Word oraz ASCII pod następującymi adresami:

- http://futureos.cpc-live.com/files/FutureOS_Manual.doc
- http://futureos.cpc-live.com/files/FutureOS_Manual.zip
- http://futureos.cpc-live.com/files/FutureOS_ROM-DOK-E.zip

Autor artykułu zaprasza osoby zainteresowane do kontaktu pod adresem mailowym: libicki@poczta.fm

Remont Commodore 64

W ostatnim czasie trafiło do sporo sprzetu retro. mnie Wśród starych pecetów odnalazłem cały zestaw Ζ Commodore 64, choć nieco zdekompletowany. Musiałem więc uzupełnić kilka części niektóre miałem, а inne dokupiłem na słynnym portalu aukcyjnym. Cała sprawa zajęła mi trochę czasu i pomyślałem sobie, że na łamach naszego pisma pokażę, jak można poradzić sobie z remontem Commodore bez ponoszenia dużych nakładów finansowych.

Zacznijmy od tego, że sprzęt po odnalezieniu nie prezentował się najlepiej. Komputer nie był skręcony, obudowa miała kilka ubytków, a na płcie głównej brakowało kilku układów, w tym oczywiście SID-a. Lepiej wyglądała stacja dyskietek 1541-II, a wraz z nia odnalazło się sporo dyskietek, w tym nawet kilka oryginalnych gier. Niestety nie miałem przewodu Serial, ale to żaden problem. Najciekawszym znaleziskiem był dla mnie tajemniczy cartridge bez obudowy, który koniec końców okazał się zwykłym Black Boxem.

Nie jestem zapalonym zbieraczem komputerów, dlatego nie raz ogarnia mnie pokusa pozbycia się zbiorów,

które może się są specjalnie okazałe, ale starannie wybrane i budzą we mnie wielki sentyment. Czasem myślę, czy nie zamienić całego sprzętu na emulator, urządzenie typu Mist czy Malinka, ale uruchamianie gier czy dem na "udawanym" sprzęcie jest dla mnie niewystarczające. Dlatego nowo zdobyty zestaw w C64 postanowiłem zatrzymać, mimo że mam już na biurku lekko rozbudowaną Komodę z cartridgem EasyFlash i interfejsem SD2IEC.

Na pierwszy ogień rozkręciłem komputer i stwierdziłem, że w środku jest oczywiście pełno kurzu, ale obudowa jest w bardzo dobrym stanie. Płyta główna nie ma układu SID, ale na szczęście posiadałem nadprogramowy układ wymontowany z uszkodzonego C64 wiele lat temu. Biorąc pod uwagę popularność ofert tych układów na portalach aukcyjnych i absurdalne ceny(nawet ok. 100 zł z wysyłką) nie mogłem trafić lepiej.

Miałem tylko obawę, że uszkodzona będzie płyta główna, ale nie znalazłem na niej uszkodzeń widocznych gołym okiem. Drżącą ręką podłączyłem zasilacz i okazało się, że komputer normalnie działa. Troche bardziei problematyczna okazała się klawiatura, bo przy naciśnięciu dowolnego klawisza trzeba było użyć bardzo wielkiej siły, ale litera pojawiła się na ekranie. Już myślałem, że jest to problem mechaniczny typu zniszczona folia klawiatury, a rozwiązanie okazało się banalne.



STREFA COMMODORE

Rozkręciłem klawiaturę, po czym od razu zauważyłem, że ktoś ją źle złożył, poza tym była bardzo mocno zabrudzona. Z zewnątrz nie było to widoczne, ale wyglądało to tak, jakby ktoś składać całość "na szybko" i głównie sprawdził, czy wszystkie klawisze trzymają się na swoich miejscach.

voich miejscach.

 Image: State of the state

niewiele to dało. Na domiar złego, skalowanie jest mocno widoczne i nie podoba mi się pikseloza na literach.

Telewizor Flatron zachował się zdecydowanie lepiej, zobaczyłem kolory i całkiem dobrą jakość, ale nijak nie mogłem uzyskać ekranu w trybie Progressive. Ciągle uparcie aktywował się tryb Interlace, bo przeszkadza jeszcze bardziej niż nawet brzydkie skalowanie. Dla pewności, pod ten sam wyświetlacz podłączyłem drugi C64 oraz Amigę 500 - w obu przypadkach z tym samym rezultatem. W Internecie wiele osób poleca telewizory Flatron z tej serii, ale jak się okazuje - zależy na jaki egzemplarz trafimy. Trzeba uważać.

Cóż mogę powiedzieć - wygrał zdecydowanie Sharp. Jest to telewizor nietypowy, bo nie posiada w ogóle możliwości podłączenia przez gniazdo VGA. Ma za to opcję S-Video oraz Scart z obsługą RGB, a jego rozdzielczość to 640x400 pikseli. I chyba ta ostatnia cecha - niska rozdzielczość spowodowała, że obraz wygląda najlepiej. Nie widać charakterystycznego skalowania, a ekran wygląda bardziej jak na wyświetlaczu CRT. Oczywiście nie jest to dokładnie ten sam efekt, ale

Oddzielną kwestią było wyczyszczenie jednostki, bo klawiatura wewnątrz miała jeszcze więcej włosów i kurzu niż na płycie głównej. Nie wiem jak to biorąc szczególnie możliwe, pod uwagę, że klawiatura jest jednak elementem, gdzie nie ma zbyt wiele miejsca, ale fakt, że po wyczyszczeniu i ostrożnym skręceniu wszystkie klawisze zaczęły "magicznie" działać lekko i bez zacięć. "Good for you", jak to mówią w amerykańskiej edycji programu Mam Talent.

Skoro już miałem działający kolejny egzemplarz C64, postanowiłem wypróbować jego działanie na kilku telewizorach LCD jakie udało mi się zdobyć w międzyczasie. Głównie chodziło mi o to, czy jakość obrazu będzie rózniła się w stosunku do modelu G, który stoi na moim biurku od lat. Poza tym nie miałem nigdy okazji mieć 5 monitorów obok siebie, a teraz nadarzyła się okazja. Dlatego podłączyłem oba Commodore do następujących modeli:

- Samsung 242MP,
- Samsung 940MG,
- Samsung 932MP,
- LG Flatron M1721A,
- Sharp 20SH1E.

Pewnie spytacie: dlaczego tyle Samsungów? Po prostu akurat takie telewizory miałem na miejscu. Szybko jednak okazało się, że wszystkie 3 modele koreańskiej firmy mają te samą przypadłość. Mianowicie po podłączeniu obrazu przez złącze S-Video... nie ma koloru! Obraz wygląda tak, jakby sygnał był zbyt słaby, bo co jakiś czas następuje drganie obrazu, ale wszystko jest nadal czarno-białe. Próbowałem zmieniać ustawienia w menu, niestety





dla mnie jest to kompromis pomiędzy kineskopem, który męczy wzrok a LCD, gdzie filtry "poprawiające" obraz powodują, że obraz nie wygląda dobrze - piksele są zbyt rozciągnięte i za bardzo wyróżniają się na mniej kontrastowym tle.

Problem samego komputera mamy rozwiązany. Wyczyściłem dokładnie obudowę i mój nowy C64 wygląda prawie jak nowy. Przejdźmy teraz do stacji dyskietek, bo tutaj pojawiły się problemy związane z obsługą SD2IEC. Mianowicie wymyśliłem sobie, ze podłączę oba urządzenia jednocześnie, aby łatwiej sprawdzić odczyt i zapis na dyskietkach.

Muszę przyznać jedno - stacja była w dobrym stanie wizualnym, ale nie spodziewałem się, że wszystko od razu zadziała. A tutaj niespodzianka, bo nie dość, że napęd odczytuje i zapisuje dyskietki sprzed 30 lat, to jeszcze pracuje cicho i bezawaryjnie. Nie musiałem robić praktycznie nic, za wyjątkiem czyszczenia oraz kłopotów wynikłych z moim interfejsem SD2IEC, ale to zupełnie inna sprawa.

Interfejs miałem podłączony na zewnątrz, w formie płytki z przelotowym gniazdem Serial. Niestety okazało się, że mimo teoretycznie prawidłowego ustawienia numerów urządzeń jako 8 i 9, transmisja danych ze stacji jest zakłócana przez SD2IEC. Po prostu dane były odczytywane, ale z dziwnymi przestojami i błędami. Zamieniłem więc płytkę na wersję, którą można zamontować wewnątrz obudowy i problemy zniknęły. Oczywiście musiałem przylutować przewody do płyty głównej Komodorka, ale nie jest to tak bardzo skomplikowane, a dzięki temu stacja i interfejs pracują razem bez najmniejszych problemów.

Kłopotliwe jest tylko nagrywanie dyskietek z plików w formacie D64, a to ze względu na brak przycisku Reset w SD2IEC. Poprzednia wersja miała możliwość resetowania, a więc programy nie posiadające możliwości odczytywania danych z innych katalogów niż aktualnie ustawiony, można było łatwo wykorzystać po prostu zapisując ważne pliki w katalogu głównym.

Tutaj brak jest takiej możliwości, tak więc program do kopiowania CBM-Command musiałem najpierw nagrać na oddzielnej dyskietce. W przeciwnym wypadku nie mogłem odczytać żadnego pliku D64, bo program odczytywał dane od razu z wnętrza obrazu dyskietki. Przycisk Reset można dolutować do mojej płytki SD2IEC i chcę to zrobić w najbliższym czasie, ale moja rada jest taka - jeśli kupujecie podobny interfejs sprawdźcie, czy ma przyciski do zmiany obrazów dyskietek. Zaoszczędzi Wam to czasu i nerwów.

Cartridge Black Box, o którym napisałem wcześniej był dla mnie zagadką, bo właściwie nie bardzo wiedziałem, czym jest. Dostałem płytkę z przylutowanym przyciskiem na końcu dziwnego metalowego elementu i przyznam, że miałem watpliwości, czy w ogóle jest to płytka, którą warto sprawdzać. Znalazłem jednak dużo zdjęć podobnych modułów w sieci i uznałem, że powinno być "dobrze". Cartridge włożony pierwszy raz nie uruchomił się, ale po wyczyszczeniu i poprawieniu kilku styków dotknięciem goracej lutownicy, zobaczyłem ekran kontrolny z napisem "BASIC IM-PROVED BY BLACK BOX". Pełny sukces.

W moim zestawie jest jeszcze oryginalny magnetofon oraz drukarka Commodore MPS 1230. Do tej pory jeszcze ich nie sprawdziłem, bo nie mam żadnej kasety, natomiast do drukarki brakuje mi taśmy barwiącej. Magnetofon mam w oryginalnym opakowaniu, z fabrycznymi zabezpieczeniami ze styropianu, więc ma dla mnie wartość kolekcjonerską, niezależnie od sprawności. Będę chciał go jednak rozebrać, wyczyścić, wymienić gumki w środku i kupić kilka kaset, aby poczuć prawdziwy klimat retro. W końcu mój pierwszy C64 był właśnie tylko z magnetofonem, bo nie było mnie stać od razu na stację dyskietek.

Nie omieszkam się podzielić z Wami kolejnymi wrażeniami, a już szykuje mi się nowa paczka ze sprzętem. Żegnam się do następnego razu i osobiście bardzo polecam zajęcie się renowacja sprzętu retro. Chciałbym też w międzyczasie wybielić kilka moich obudów, głównie od C64 i Amigi, tak więc będzie o czym pisać.

> Opracował: Leszek Rosiński



다리

Instrukcja USR w przykładach

Czy zastanawialiście się kiedyś, jak użyć polecenia USR w Basicu? W tym artykule omówię tę przydatną funkcję razem z przykładami dla 8-bitowych komputerów Commodore. Przeanalizujemy także użycie liczb pomiędzy Basiciem i kodem maszynowym. Zapraszam.

Zarówno polecenia USR, jak i SYS są podobne do instrukcji GOSUB występującej w każdym dialekcie Basica. Jednak zamiast przesyłać sterowanie programu do podprogramu można spowodować, że kontrola przechodzi do podprogramu języka maszynowego. Tak właśnie działają instrukcje USR i SYS. Pierwsza z nich ma dodatkową możliwość przesyłania liczb lub informacji także do podprogramu języka maszynowego.

Ogólny format polecenia USR to:

n=USR(v)

gdzie N oznacza dowolną nazwę zmiennej, a V jest zmienną, której wartość ma zostać przeniesiona do podprogramu Asemblera. Po powrocie do Basica, podprogram języka maszynowego umieści nowo obliczoną wartość w zmiennej N. Przekazywanie wartości i informacji pomiędzy jednym i drugim językiem odbywa się za pośrednictwem akumulatora zmiennoprzecinkowego, czyli FAC. Technicznie rzecz biorąc, jest to pięć kolejnych bajtów w pamięci, które są używane do przechowywania liczb zmiennoprzecinkowych. Informacja adresowa dla podprogramu języka maszynowego jest określona w adresach 785 i 786 w Commodore 64 i jest zapisywana w standardowym formacie LBHB (czyli Low Byte, High Byte).

Na przykład, poniższe polecenie:

nv=USR(ov)

w programie Basica najpierw przesyła wartość OV do FAC. Następnie program rozgałęzia się do podprogramu języka maszynowego, którego adres początkowy jest przechowywany w adresach 785 i 786. Przed opuszczeniem podprogramu programista przechowuje nowo obliczoną wartość w FAC i wydaje polecenie RTS powodujące powrót z podprogramu. Po przywróceniu Basica, wartość w FAC jest utrzymywana przez zmienną NV, a działanie programu w Ba-

sicu jest kontynuowane od miejsca, w którym został wcześniej przerwany.

Jedyną rzeczą uniemożliwiającą pełne wykorzystanie funkcji USR jest konwersja danych zmiennoprzecinkowych w FAC. Liczba musi najpierw być w formacie używanym przez program języka maszynowego, a następnie obliczona wartość musi zostać ponownie sformatowana do FAC, aby możliwy był powrót do programu w Basicu.

Cała sztuka polega na znajomości położenia dwóch ważnych podprogramów w pamięci. Jeden z nich konwertuje zawartość FAC na liczbę całkowitą o podwójnej precyzji, przechowywaną w adresach \$61 i \$62 w komputerze PET oraz odpowiednio -\$64 i \$65 on w VIC-20 oraz Commodore 64.

Poniższe podprogramy są bardzo łatwe w użyciu. Na początek zmień zawartość Basica według listingów przedstawionych jako Program numer 1 oraz Program numer 2. Pierwszy podprogram pozwala dokonać konwersji zawartości FAC na liczbę o podwójnej precyzji:

JSR \$DBA7 LDA \$61

Рг	ogr	ам	пг	1:	Wersja dla PET
03 03	3C 44	20 62	A7 85	DB FB	A5 61 85 PC A5 A0 00 B1 FB A8
03	4C	A9	00	20	GaD D2 60 00 FF
Pr	ogr	ам	пг	2:	Wersja dla BASIC 4.0
03	3C	20	D1	CD	A5 61 85 FC A5
03	44	62	85	FB	A0 00 B1 FB A8
03	4C	A9	00	20	BC C4 60 00 FF



LDY \$62

Rejestry A i Y zawierają skonwertowaną wartość całkowitą T w równaniu zapisanym w Basicu jako:

S=USR(T)

Kiedy musisz przenieść wartość z programu języka maszynowego z powrotem do Basica, liczba całkowita musi zostać umieszczona w FAC w odpowiednim formacie. Poniższy kod spełni to zadanie bardzo dobrze:

LDA	\$61
LDY	\$62
JSR	\$D26D
RTS	

Zmienna S będzie zawierać przeliczoną wartość dostarczoną przez program języka maszynowego. Te dwa podprogramy powinny wystarczyć w przypadku większości programów.

Aby zilustrować użycie tych programów, użyjmy funkcji USR do symulacji działania instrukcji PEEK. Dzięki temu możemy ocenić podprogramy, porównując wyniki testu z wynikami ogólnie znanej instrukcji. Wybrany przykład jest co prawda przydatny główne jako narzędzie do nauki, ale w końcu każdy kiedyś zaczynał programować.

Właściwy program dla VIC-20 to Program numer 3, natomiast wersja dla C64 - Program numer 4. Oba znajdziecie w ramkach obok. Dalej trzeba wpisać Program numer 5, zwracając szczególną uwagę na linię 120. Liczby w liniach zawierających instrukcje DATA zawierają program języka maszynowego i muszą być wpisane bardzo dokładnie. Uwaga: przed uruchomieniem programu koniecznie go zapisz na taśmie lub dyskietce.

W języku maszynowym będzie to wyglądało tak:

Program nr 3: Wersja dla VIC-20

10 FOR A = 828 TO 849 : READ D : POKE A, D : NEXT 20 DATA 32, 155, 220, 165, 100, 133, 252, 165 30 DATA 101, 133, 251, 160, 0, 177, 251, 168 40 DATA 169, 0, 32, 145, 211, 96

Program nr 4: Wersja dla Commodore 64

10 FOR A = 828 TO 849 : READ D : POKE A, D : NEXT 20 DATA 32, 155, 188, 165, 100, 133, 252, 165 30 DATA 101, 133, 251, 160, 0, 177, 251, 168 40 DATA 169, 0, 32, 145, 179, 96

\$033C	20 A7	DB	JSR \$DBA7
			IN \$61, \$62
\$Ø33F	A5 61		LDA \$61
\$034i	85 FC		STA \$FC
\$0343	A5 62		LDA \$62
\$0345	85 FB		STA \$FB
\$0347	AØ 00		LDY #\$00
\$0349	B1 FB		LDA (\$FB),Y
			ADDR
\$034B	A8		TAY
\$034C	A9 00		LDA #\$00
\$034E	20 GD	D2	JSR \$D26D
\$0351	60		RTS

Teraz spójrzmy na wszystkie siedem etapów, które sa niezbędne do funkcjonowania funkcji USR. Pierkrokiem jest umieszczenie wszym początkowego adresu języka maszynowego w adresach \$01 i \$02. Najmniej znaczący bajt adresu (LSB) jest przechowywany w pierwszym adresie, a najbardziej znaczący bajt adresu (MSB) jest przechowywany w adresie drugim. Adres startowy to \$033C, dlatego adres \$00 powinien zawierać następujący kod:

\$0000 4C 3C 03 JMP \$033C

W języku BASIC wartości adresowe muszą być podane w systemie dziesiętnym. Nie ma potrzeby wywoływania instrukcji POKE z adresem \$310 (wersja dla C64), ponieważ została zainicjowana do właściwej wartości po włączeniu komputera. W swoim programie w Basicu ustal wartość T w równaniu S=USR(T) Aby korzystać z podprogramów podanych w tym artykule, T musi być liczbą całkowitą o wartości od 0 do 65535. Jest to pełny zakres 2-bajtowych liczb całkowitych, a w systemie szesnastkowym jest równy od \$0000 do \$FFFF. W tym przykładzie linie od 130 do 150 są używane do wprowadzania i testowania liczby całkowitej z tego samego zakresu.

Gdy wykonywane jest S=USR(T), jest to równoważne instrukcji SYS (828). W obu przypadkach kontrola jest przekazywana z Basica do programu języka maszynowego. Funkcja USR różni się od polecenia SYS tym, że FAC może służyć do przekazywania rzeczywistych danych do i z programu w Asemblerze.

W programie języka maszynowego konwertujemy wartość z FAC na dwubajtową liczbę całkowitą. Instrukcje zapisane w adresach od \$033F do \$034A są używane do pobierania wartości, którą chcemy przekazać do Basica.

Instrukcje od adresów \$034B do \$0351 wpisują wartość całkowitą w rejestrach A i Y. Ponieważ symulowana wartość PEEK jest tak naprawdę tylko jedną liczbą całkowitą, MSB jest ustawione na zero. JSR w \$D26D zamieni wartości w rejestrach A i Y na zmiennoprzecinkowe i umieści je w FAC. Wreszcie, RTS zwróci kontrolę do programu w Basicu. 30 STREFA COMMODORE

Program numer 5

100 REM SAVE BEFORE RUNNING 110 POKE 1, 60 : POKE 2, 3 : REM JMP \$033C 120 REM FOR C-64, USE POKE 785, 60 : POKE 786, 3 130 PRINT "SIMULATED PEEK" : PRINT 140 PRINT "INPUT AN ADDRESS BETWEEN 0 AND " 65535" 150 INPUT T : IF T < 0 OR T > 65535 OR INT(T) <> T THEN 140 160 S = USR(T) : REM SYS 828 (\$003C) 170 PRINT S" = PEEK("T")" , PEEK(T) : PRINT 180 GOTO 140

Prawdziwa zmienna S zostanie teraz przypisana wartości umieszczonej w FAC przez program języka maszynowego. Linie 170 i 180 w Programie numer 5 wyświetlają zarówno wartość S, jak i rzeczywistą wartość PEEK w celu sprawdzenia, czy symulacja jest prawidłowa.

To wszystko, co trzeba wiedzieć. Na koniec dodam kilka słów na temat przechowywania danych w niskich (Low Byte) i wysokich (High Byte) bajtach. Wcześniej wspomniałem o metodzie LBHB, która właśnie używa tych pojęć. Jest to sposób zapisywania dużych liczb stosowany przez wiele mikrokomputerów.

Bajt może zmieścić liczbę nie większą niż 255, dlatego do reprezentowania liczb większych niż 255 potrzebujemy dwóch lub więcej kolejnych bajtów. Format LBHB obejmuje metodę, w której liczby są dzielone, a następnie zapisywane w pamięci jako najmniej znaczącym bajt (LSB) najpierw, a następnie jako najbardziej znaczący bajt (MSB).

Liczba od 256 do 65535 jest przechowywana w pamięci RAM za pomocą dwóch kolejnych bajtów. Drugi bajt (czyli najbardziej znaczący) otrzymuje się poprzez podzielenie oryginalnej liczby przez 256, a następnie zapisanie wartości liczby całkowitej (bez ułamków) w MSB. Pozostała część tego podziału jest przechowywana w pierwszym (czyli najmniej znaczącym) bajcie. Można powiedzieć, że korzystamy z poniższego wzoru:

liczba = LSB + (MSB \times 256)

Na przykład, powiedzmy, że chcemy skierować działanie do adresu 828, czyli bufora magnetofonu. W tym celu należy umieścić wartość 848 a adresach \$01 i \$02. Musimy cały czas korzystać z formatu LBHB. Jak to zrobić? Dzielimy 828 przez 256 i zapisujemy wynik w drugim bajcie. Pozostałą część z podziału zapisujemy w bajcie pierwszym. Dalej używamy wzoru:

N = bajt pierwszy + (256 × bajt drugi)

Zmienna N oznacza oczywiście naszą liczbę, a całość pozwala odczytać liczby w formacie LBHB. Drugi sposób to wzór w innej postaci:

NN = INT (N / 256): POKE bajt pierwszy, N - (NN ★ 256): POKE bajt drugi, NN

Technika ta wymaga kilku prób, aby w pełni zrozumieć działanie oraz zrozumienia wzorów matematycznych, ale nie ominiemy tego, jeśli chcemy programować w języku maszynowym. W zamian uzyskujemy dużo większe możliwości używania podprogramów.

> Opracował: Mariusz Wasilewski









Jeszcze raz o czyszczeniu ekranu

Poniżej przedstawiam procedurą języka maszynowego, która czyści ekran wysokiej rozdzielczości Commodore 64 w mniej niż sekundę.

Jeśli kiedykolwiek używałeś grafiki o wysokiej rozdzielczości w C64, prawdopodobnie zauważyłeś, ile czasu zajmuje wymazanie zawartości ekranu w Basicu. Można to zrobić za pomocą instrukcji POKE zapisując adresy od "ósmego" kilobajta zerami. Wystarczy użyć poniższej pętli:

FOR J = 8192 TO 16192 : Poke J,0 : Next

Zajmuje to około 30 sekund, co nie jest bardzo długim czasem, ale może wydawać się niewystarczające, jeśli chcemy używać tej procedury dość często. W zamian możemy użyć procedury języka maszynowego, która czyści ekran w czasie krótszym niż jedna sekunda. Można go wstawić do dowolnego programu korzystającego z grafiki o wysokiej rozdzielczości.

Zwróć uwagę, że nie musisz rozumieć języka maszynowego, aby użyć tego programu. Po prostu go wpisz i uruchom. Po uruchomieniu procedury, użyj instrukcji:

SYS 828

wtedy, gdy chcesz wyczyścić ekran. Aby przetestować działanie uruchom



Tak może wyglądać ekran po przełączeniu trybu graficznego bez wyczyszczenia zawartości.

program, a następnie wpisz następujące polecenia:

POKE 53265, PEEK(53265) OR 32 : Poke 53272, PEEK(53272) OR 8

Na ekranie pojawią się "krzaki". Teraz naciśnij kombinację klawiszy SHIFT + CLR/HOME i wprowadź SYS 828. Ekran zostanie wyczyszczony w mgnieniu oka.

Zazwyczaj będziesz chciał zapełnić ekran wartościami zerowymi, aby stał się pusty. Tej samej procedury możesz jednak użyć także po to, aby wypełnić ekran Hires dowolną wartością od 0 do 255. Wystarczy zmienić drugą liczbę w trzeciej instrukcji DATA z 0 na inną. Procedura jest relokowalna. Aby zmienić lokalizację, zmień zmienną SA w linii 10 z 828 na inny bezpieczny adres (na przykład 49152).

Zmienna HS w wierszu 10 to adres początkowy ekranu Hires. Jeśli użyjesz adresu innego niż 8192, pamiętaj, aby zmienić także wartość zmiennej HS.

> Opracował: Marcin Libicki

```
10 SA=828:HS=8192:POKE2,HS-256*INT(HS/256):
POKE3,INT(HS/256)
20 FOR A=SATOSA+31:READB:POKEA, B:NEXT
30 DATA 165,2,133,4,165,3
40 DATA 133,5,162,30,160,0
50 DATA 169,0,145,4,136,208
60 DATA 251,230,5,202,16,242
70 DATA 160,64,145,4,136,16
80 DATA 251,96
```



Szybkie przewijanie ekranu może być denerwujące, szczególnie po użyciu polecenia LIST. Na szczęście w prosty sposób można kontrolować te prędkość i w razie potrzeby spowolnić, przyspieszyć lub zatrzymać przesuwanie informacji.

Jeśli nie masz drukarki, edycja programu może czasami wymagać specjalnych technik, aby zorientować się w budowie całego listingu. Podczas wykonywania instrukcji LIST, zawartość programu jest przewijana zbyt szybko, aby można było przeczytać poszczególne linie. Istnieją programy języka maszynowego, które pozwalają spowolnić działanie, jednak jest też duzo łatwiejsza i krótsza technika. W tym wypadku kluczowy jest adres pamięci 37879.

Podręcznik dołączany do VIC-20 nie omawia tej możliwości. Trzeba jednak powiedzieć, że niezależnie od tego, czy było to planowane przez konstruktorów komputera, zmiana wartości pod wspomnianym adresem powoduję zmianę szybkości zegara systemowego. Może on działać wolniej lub szybciej niż w czasie rzeczywistym.

Aby zrozumieć wprowadzane zmiany, najlepiej na początku do komórki pamięci 37879 wprowadzić wartość 0. Spowoduje to, że zegar będzie działał 60 razy szybciej niż w czasie rzeczywistym. Robimy to za pomocą typowej instrukcji POKE, czyli wpisujemy POKE 37879,0. Analogicznie, jeśli wprowadzimy linię POKE 37879,255 - zegar będzie utrzymywał czas na około poziomie ok. 25% czasu rzeczywistego. Normalna wartość tej lokalizacji to 64, dlatego w celu przywrócenia domyślnej szybkości trzeba użyć wpisu POKE 37879,64.

Wydaje mi się, że funkcję tę można wykorzystać najlepiej w przypadku instrukcji LIST. Po wpisaniu POKE 37879,0 i ponownej próbie wyświetlenia listy, będziemy mieli znacznie lepszą kontrolę nad przewijaniem ekranu. Jeśli przytrzymamy klawisz SHIFT podczas wyświetlania, komputer wyświetli jedną linię co ok. pół sekundy.

Przytrzymanie klawisza CTRL również spowoduje zatrzymanie listy. Użycie tych klawiszy nie zmienia faktu, że nadal możemy zatrzymać przewijanie za pomocą RUN/STOP i edytować listing jak zwykle.

Użytkownicy cartridge'a Super Expander muszą wziąć pod uwagę, że wraz ze spowolnieniem przewijania ekranu, to samo stanie się z operacjami graficznymi wykonywanymi po wpisaniu poleceń takich jak DRAW czy COLOR. Będą one działały bardzo powoli po przytrzymaniu klawisza CTRL. Funkcja ta może być jednak bardzo przydatna podczas programowania gier, gdy chcemy sprawdzić funkcje rysujące grafikę.

Wraz z wprowadzeniem instrukcji POKE 37879 zmienia się szybkość ruchu kursora. Jego szybkość będzie większa, tak więc ustawienie kursora w odpowiednim miejscu na ekranie może być trudne. Przy tej warto zauważyć, że jeśli okazji, wpiszemy polecenie PEEK 37879, często także zwracana jest wartość zerowa. Będzie to występować często po naciśnięciu klawiszy RUN/STOP i RE-STORE, gdy zegar będzie działał w czasie rzeczywistym.

Jeśli jednak wpiszemy linię POKE 37879,0, kursor zacznie szybko migać, mimo że nie zmieniliśmy wartości pod adresem 37879 - cały czas jest tam "zero". Tak się dzieje, bo komputer reaguje na zapisanie komórki pamięci, która wcześniej była traktowana jako "pusta", w związku z tym nie powodowała zmiany szybkości pracy.

Jak widać, opisana technika nie jest idealnym sposobem na wolniejszą lub szybszą edycję programu, ale z pewnością z wielu wypadkach będzie przydatna. Proponuję też uruchamiać różne funkcje i polecenia Basica przy jednoczesnym manipulowaniu wartością pod adresem 37879. W wielu wypadkach może to dać nieoczekiwane rezultaty, mimo że w teorii zmiany powinny mieć wpływ tylko na niektóre operacje.

Testowanie dźwięku SID

Możliwości dźwiekowe Commodore 64 sa dużo szersze niż wcześniej produkowanych komputerów, też 8-bitowych. Jeszcze wcześniej różnorodne kształty fal, manipulowanie takimi efektami jak czas narastania czy opadania dźwięku były dostępne tylko drogich syntezatorach. w Dzisiaj jesteśmy przyzwyczajeni, że komputery po prostu odtwarzają dźwięk wcześniej zapisany w formie cyfrowej. Dlatego zrozumienie ustawień układu SID może bvć kłopotliwe zarówno dla poczatkujących, jak i nieco bardziej zaawansowanych programistów, którzy chcą poznawać kolejne platformy retro.

Przedstawiony poniżej program umożliwia eksperymentowanie funkcjami dźwiękowymi, a następnie odsłuchanie rezultatów i wprowadzanie zmian. Możemy używać do ośmiu oktaw, a także korzystać z generatora szumów. Dane mogą być wyświetlane w dowolnym momencie przed rozpoczęciem przeglądania innych ustawień.

Po uruchomieniu programu pojawi się lista bieżących wartości z kursorem. Jeśli chcemy zmienić wartość, należy

po prostu wpisać nową, a następnie nacisnąć RE-TURN. Jeśli natomiast wartość na pozostać bez zmian, po prostu naciskamy RETURN.

Litery T, S i P odnoszą się do przebiegów trójkątnych, piłokształtnych i prostokątnych. N oznacza generator szumów, a za pomocą Q wychodzimy z programu. Za każdym razem, gdy chcemy zobaczyć wprowadzone dane, trzeba nacisnąć klawisz D.

Po wprowadzeniu ostatniej wartości zostanie odtworzona melodia z aktualnymi wartościami ADSR (ang. Attack, Decay, Sustain, Release), po których program zostanie zapętlony.



```
5 DIM SO (16, 8)
10 HF = 54273 : LF = 54272 : AD = 54277 : SR = 54278 : W = 54276 : U = 54296
: HP = 54275 : LP = 54274
15 FORI = 1T08 : A(I) = 0 : NEXT : W$ = "S"
20 FOR 0 = 1 T0 8 : FOR N = 1T016 : READ SO (N, 0) : NEXT : NEXT
25 FOR N = 1 T0 8 : READ D (N) : NEXT
100 PRINT "{CLEAR}" ; CHR$ (18) ; "PULSE SETTING USED ONLY WITH PULSE WAVE "
; CHR$(146)
102 PRINT
110 PRINT "WAVEFORM (T, S, P, N) = " ;W$
112 PRINT "UOLUME (1-15) = " ; A (1)
114 PRINT "OCTAVE (1-8) = " ; A(2)
116 PRINT "ATTACK SETTING (0-15) = " ; A(3)
118 PRINT "DECAY SETTING (0-15) = " ; A(5)
```

```
122 PRINT "RELEASE SETTING (0-15) = " ; A(6)
124 PRINT "HIGH PULSE SETTING (0-15) = " ; A(7)
126 PRINT "LOW PULSE SETTING (0-255) = " ; A(8)
130 PRINT "{HOME}" ; "{03 DOWN}" ;
140 PRINTTAB(33) ; "Q" GOSUB500 : IF Z$ = "D" THEN 600
150 IF Z$ = "Q" THEN PRINT "{CLEAR}" : END
155 IF Z$ ( ) CHR$ (13) THEN W$ = Z$
160 FORO = 1 T0 6
170 PRINTTAB(33) ; "?" ; : GOSUB500 : IF Z$ < > CHR$(13) THENA(0) = VAL(Z$)
180 NEXT
190 IF W$ < > "P" THEN 230
200 \text{ FORO} = 708
210 PRINTTAB(33) ; "?" ; :GOSUB500 : IF Z$ < > CHR$(13) THEN A(0) = UAL(Z$)
220 NEXT
230 POKEV, A(1)
240 POKEAD, 16 * A(3) + A(4) : POKESR, 16 * A(5) + A(6)
250 IF W$ = "T" THEN POKE W, 17
260 IF W$ = "S" THEN POKE W, 33
270 IF W$ = "P" THEN POKE W, 65 : POKEHP, A(7) : POKELP, A(8)
280 IF W$ = "N" THEN POKE W, 129
300 0 = 0
310 FOR I = 1T015 STEP2 : POKEHF, SO (I, A(2)) : POKELF, SO (I + 1, A(2)) :
0 = 0 + 1 ; FORN = 1 TOD(0)
311 NEXT
315 POKEHF, 0 : POKELF, 0 : NEXT : POKEW, 0 : POKEAD, 0 : POKESR, 0 :
POKEHP, 0 : POKELP, 0
320 GOTO 100
500 Z$ = " "
510 GETY$ : PRINTCHR$ (18) ; " " ; CHR$ (146) ; : FOR I =1T025 : NEXT :
PRINTCHR$(157) ; " " ;
515 PRINTCHR$(157); : FORI = 1T025 : NEXT : IFY$ = " " THEN510
520 PRINTY$;
530 IF Y$ = CHR$(13) THENIF LEN(Z$) = 0 THENZ$ = CHR$(13)
540 IF Y$ = CHR$(13) THEN RETURN
550 Z$ = Z$ + Y$ : GOTO510
600 PRINT "{CLEAR}"
610 PRINT : PRINT "ATTACK/DECAY = " ;16 * A(3) + A(4)
620 PRINT "SUSTAIN/RELEASE = " ;16 * A(5) + A(6)
630 PRINT : PRINT : PRINT "HIT ANY KEY TO CONTINUE" ;
640 GETY$ : IFY$ = " " THEN640
650 GOTO100
1000 DATA 1, 155, 1, 90, 1, 110, 1, 155, 1, 90, 1, 110, 1, 155, 1, 205
1010 DATA 3, 54, 2, 179, 2, 220, 3, 54, 2, 179, 2, 220, 3, 54, 3, 155
1020 DATA 6, 108, 5, 103, 5, 185, 6, 108, 5, 103, 5, 185, 6, 108, 7, 53
1030 DATA 12, 216, 10, 205, 11, 114, 12, 216, 10, 205, 11, 114, 12, 216, 14,
107
1040 DATA 25, 177, 21, 154, 22, 227, 25, 177, 21, 154, 22, 227, 25, 177, 28,
214
1050 DATA 51, 97, 43, 52, 45, 198, 51, 97, 43, 52, 45, 198, 51, 97, 57, 172
1060 DATA 102, 194, 86, 105, 91, 140, 102, 194, 86, 105, 91, 140, 102, 194,
115, 88
1070 DATA 205, 133, 172, 210, 183, 25, 205, 133, 172, 210, 183, 25, 205,
133, 230, 176
1080 DATA 500, 250, 250, 250, 250, 250, 250, 1000
```



Atari Light Pen

Jedną z najwspanialszych rzeczy w komputerach domowych, szczególnie w Atari, jest ich pełnokolorowa grafika. Niestety jednocześnie jedną z najbardziej frustrujących rzeczy w komputerach domowych może być złożoność programowania tej samej grafiki. Jak zacząć, aby nie zniechęcić się po kilku chwilach?

Jeśli jesteś ambitny, możesz zacząć czytać książki i czasopisma, które dostępne są w sieci, ale będzie to wymagać wielu godzin nauki, których zwykle dzisiaj nie mamy. Są zdecydowanie lepsze sposoby, ale stworzyć pierwszą prostą grafikę. Można przykładowo podłączyć urządzenie typu tabliczka graficzna czy nawet myszka, wtedy zdecydowanie łatwiej można uzyskać założone efekty. Programy do rysowania grafiki używają joysticków lub wiosełek (paddle) jako urządzeń wejściowych. Nie są to tradycyjne narzędzia do rysowania, ale z drugiej strony nie są trudne do opanowania, szczególnie jeśli mamy wprawę w grach. Mimo wszystko byłoby miło, gdybyśmy mogli rysować na ekranie bardziej naturalnie, bezpośrednio za pomocą jakiegoś urządzenia, które pozwala odwzorować ruchy ręki w komputerze.

W tym miejscu dochodzimy do dwóch rozwiązań. Jednym z nich jest tablet, o którym już wspomniałem, natomiast drugim - tytułowe pióro świetlne. Tak naprawdę są one bardzo podobnie zbudowane, bowiem oba tłumaczą pozycję rysika na współrzędne, które komputer rozumie jako punkt na ekranie. Dlatego oba urządzenia pozwalają ominąć etap programowania grafiki i zamienić go na "gesty".



Mimo zbliżonej zasady działania, obsługę pióra świetlnego będziemy odczuwać zupełnie inaczej niż tabliczki graficznej.

Ostateczny wybór zależy oczywiście od osobistych preferencji, ja jednak zdecydowanie wolę to pierwsze, o ile jest możliwość użycia w moich ulubionych programach. Różnice można porównać do rysowania kredą lub piórem i atramentem. Najlepiej jest spróbować swoich sił na obu urządzeniach przed podjęciem decyzji.

W moich rękach swojego czasu znalazł sie produkt firmy Atari o nazwie Light Pen, co pozwoliło mi zastanowić się głębiej nad możliwymi zastosowaniami tego nietypowego kontrolera. Zestaw Atari Light Pen zawiera rysik, który podłącza się do portu joysticka za pomocą zwykłego przewodu. Dodatkowo mamy dołączone oprogramowanie o nazwie Atari Graphics jako ROM oraz 20-stronicowy podręcznik. Rzecz jasne, teraz mówię o oryginalnym wyposażeniu.

Pakiet działa na każdym komputerze Atari, ale zalecane jest co najmniej 48 kilobajtów pamięci RAM, co jest wymagane do zapisywania obrazów na dysku. Zapisywanie na taśmie wymaga mniejszej ilości pamięci, w tym wypadku wystarcza nawet 16 kilobajtów RAM. Myślę, że oprogramowanie mogłoby być pod tym względem lepiej zoptymalizowane, ale takie wymagania były kiedyś podawane przez producenta.

Drugim urządzenie, którego co prawda nie miałem jest Atari Touch Tablet tablet graficzny z dwoma przyciskami,

36 STREFA ATARI



który podłącza się do także portu joysticka. W komplecie był też plastikowy rysik z małym przyciskiem , który podłącza się do tabletu oraz oprogramowanie AtariArtist (również ROM), dyskietka instalacyjna i podobny jak wcześniej podręcznik. Pakiet powinien działać na każdym komputerze Atari z co najmniej 16 kilobajtami pamięci.

Pierwsze kroki na obu urządzeniach są bardzo przyjemne. Wystarczy podłączyć pióro lub tablet, włożyć cartridge i włączyć komputer. Jeśli masz stację dyskietek, przed uruchomieniem trzeba użyć dyskietki z systemem DOS. Oba pakiety raczej nie mają problemów na różnych wersjach systemu operacyjnego, a przynajmniej nie znalazłem informacji o większych trudnościach w użytkowaniu.

Trzeba dodać, że pióra świetlne wymagają kalibracji, więc pierwszą rzeczą, o którą prosi oprogramowanie Ataritowanego przez ekran. Jeśli nie ma wystarczającej ilości światła, komputer nie może ustalić, jakie miejsce jest wskazywane przez użytkownika.

Właściwa kalibracja to pięta achillesowa tego typu rządzeń, a Atari Light Pen nie jest wyjątkiem. Skuteczność może być tu różna w zależności od ekranu, odczyty mogą być zakłócane przez inne urządzenia i generalnie trzeba mocno zwiększyć jasność, aby wszystko działało. Niestety z tego względu niektóre ciemne kolory w ogóle nie będą rejestrowane.

Jednym z rozwiązań tego problemu jest rysowanie grafiki jasnymi kolorami, a następnie zmiana ich na ciemniejsze w programie, już po zakończeniu pracy. Na szczęście jest to bardzo łatwe do zrobienia. Tablet dotykowy oczywiście nie wymaga kalibracji, więc od razu możemy zacząć używać dołączony program.



Rysowanie w programie Atari Artist.

Graphics, jest wskazywanie piórem określonych miejsc na ekranie. Jeśli nic się nie dzieje, prawdopodobnie oznacza to, że jaskrawość telewizora lub monitora jest zbyt niska. Pamiętajmy, że pióro zawiera sensor, który faktycznie rozpoznaje jasność światła emiMimo, że programy AtariGraphics (dla pióra świetlnego) i AtariArtist (dla tabletu dotykowego) mają wiele wspólnych elementów, mają także różne rodzaje menu. Widaż, że AtariGraphics było projektowane pod wpływem filozofii pracy jak na ówczesnych Macintoshach. Po lewej stronie ekranu znajdują się cztery małe, białe zakładki, a naciśnięcie pióra świetlnego na jednym z nich powoduje przesuwanie menu nad rysunkiem.

Progrsm pozwala wybierać różne tryby rysowania, kolory, wzory i opcje zapisywania danych. Gdy ponownie wskażemy zakładkę, menu zsuwa się z ekranu, pozostawiając nienaruszony rysunek. Możesz również spowodować, że karty znikną za pomocą klawiatury, naciskając klawisz TAB. W przeciwieństwie do tego menu, oprogramowanie AtariArtist ma bardziej konwencionalnv ekran. Naciśniecie przycisku na tablecie dotykowym lub przycisku na rysiku powoduje po prostu natychmiastowe przełączenie pomiędzy ekranem rysunku a menu.

W przypadku obu programów wybranie opcji menu wymaga tylko dwóch kroków. Wszystkie opcje są przedstawiane na ekranie jako ikony - są małe, ale czytelne, zatem nie wymagają wyjaśnienia. Za pomoca pióra świetlnego wskazujemy żądaną ikonę, a następnie delikatnie dociskamy pióro do ekranu. Czuła na nacisk końcówka pióra wybór użytkownika. Natomiast podczas używania tabletu, przesuwamy wskaźnik ekranie. delikatnie dotykajac po powierzchni rysikiem lub palcem. Gdy wskaźnik rozpozna żądaną funkcję, należy nacisnąć przycisk uruchamiania na tablecie lub na rysiku.

Zamiast menu wysuwanego, oprogramowanie AtariArtist posiada oddzielny ekran z menu, który jest dostępny po naciśnięciu przycisku. Należy stwierdzić, że opcje dostępne w AtariArtist i AtariGraphics są dość podobne. W każdym systemie możesz rysować odręcznie, program automatycznie rysuje okręgi, prostokąty i proste linie łączące dowolne dwa punkty, wypełnia dowolny kształt szeroką gamą kolorów i wzorów, pozwala zmieniać kolory i wzory czy też powiększać obraz. Można rysować w trybie

STREFA ATARI



dwukierunkowym lub czterokierunkowym, w którym każda zmiana w jednej części ekranu jest odbijana lustrzanie po przeciwnej stronie. Oczywiście mamy też dostępne opcje zapisywania i ładowania gotowej grafiki z dysku lub taśmy.

Mimo tych podobieństw, pomiędzy programami istnieją też istotne różnice. Na przykład AtariGraphics (dla pióra świetlnego) posiada trzy główne funkcje, których brakuje w AtariArtist. Mianowicie, możesz wprowadzać tekst na ekranie z klawiatury i nałożyć na siebie siatkę (w formie kropek) nad obrazem. Funkcja ta działa jak pomoc przy rysowaniu prostych linni lub konieczności zachowania określonych proporcji.

Z drugiej strony oprogramowanie tabletu AtariArtist ma opcje, których brakuje w AtariGraphics. Przykładowo, można rysować wypełnione koła i prostokąty, możesz rysować "promienie" w formie wielu lini wychodzących z jednego punktu, można też zmieniać grubość pędzla od cienkich linii do bardzo szerokich.

Oczywiście istnieją również ograniczenia dotyczące każdego programu, ale równie dobrze można traktować te rozwiązania jako uzupełniające, jeśli uznasz, że jest to wygodniejsze. Figury geometryczne można tworzyć za pomocą pióra świetlnego, a następne wypełniać je za pomocą tabliczki graficznej.

Proste linie można narysować za pomocą pióra i łączyć je przez oprogramowanie tabletu. Ogólnie można powiedzieć, że każda grafika, która zostanie utworzony za pomocą jednego programu może być skopiowana i obrabiana dalej w drugim.

Zarówno AtariGraphics, jak i AtariArtist umożliwiają pracę z paletą 128 kolorów, ale tylko cztery mogą być wyświetlane



Opcje wypełniania wzorami i kolorami.

jednocześnie na ekranie. Na początku może to być postrzegane jako dziwne ograniczenie, ponieważ komputery Atari mają przecież paletę 256 kolorów. Wyjaśnienie jest takie, że firma Atari musiała pójść na kompromis, aby oba pakiety były kompatybilne ze wszystkimi komputerami Atari, także wcześniejszymi modelami. Oczywiście brano pod uwagę mniejszą sprzedaż, gdyby programy działały tylko na niektórych modelach.

Ograniczenie czterech równoczesnych kolorów to kolejny "zgniły" kompromis. Niektóre tryby graficzne Atari mogą wyświetlać więcej niż cztery kolory, ale w niższych rozdzielczościach. Atari-Artist i AtariGraphics wykorzystują tryb graficzny o rozdzielczości 160x192 punktów. Jest dostępny dopiero na Atari XL, także w języku Basic. Maksymalna rozdzielczość wynosi 320x192 pikseli, ale ten tryb oferuje tylko dwa kolory.

Dlatego producent zamienił mniejszą rozdzielczość na większą ilośc kolorów, co z pewnością jest pozytywne jeśli chcemy narysować grafikę typu tło do gry czy dema, natomiast gorzej sprawdza się przy tworzeniu obrazów czysto technicznych.

Menu kolorów AtariArtist zawiera efektowną opcję, która pozwala tymczasowo zastąpić dowolny z czterech kolorów ekranu za pomocą przewijanej 128-kolorowej tęczy. Inna opcja wywołuje ekran pomocy dla początkujących. Opcja powiększania jest o wiele bardziej wszechstronna - możesz wykonać praktycznie dowolną funkcję dostępną w trybie normalnym, podczas gdy oprogramowanie pióra świetlnego pozwala jedynie zmieniać kolory pikseli.

Ponadto grafiki wykonane za pomocą tabletu zajmują dużo mniej miejsca na taśmie lub dysku niż obrazy narysowane za pomocą pióra świetlnego.

Możliwości rysowania oferowane przez programy AtariGraphics i AtariArtist są tak podobne, że decyzja o zakupie kiedyś mogła być bardzo trudna. Dzisiaj jesteśmy ograniczeni do urządzenia, które akurat uda się zdobyć, dlatego
myślę, że warto wypróbować obie opcje. W mojej kolekcji z pewnością znajdzie się zarówno pióro świetlne, jak i tablet, o ile znajdę ten drugi na portalach aukcyjnych, gdzie zwykle szukam sprzętu retro.

Dla mnie idea pióra jest bardziej naturalna, bo to jest sposób, w który zawsze rysujemy, niezależnie od stosowanego narzędzia. Natomiast tablet dotykowy może wydawać się nienaturalny, bowiem oddziela mechaniczne działanie od swobodnej techniki tworzenia dzieła.

Jednak działanie obu urządzeń może powodować pewne problemy, jak pisałem wcześniej, dlatego nie ma jednej odpowiedzi - wszystko zależy od przyzwyczajeń użytkownika.

Tablet ma dodatkową zaletę, która polega na tym, że można rysować nie tylko bezpośrednio przy komputerze, ale też z większej odległości. Mimo wszystko podczas rysowania należy patrzeć na ekran, tak samo jak podczas pisania tekstu nie obserwujemy ciągle klawiatury.

Dzisiaj, w dobie dostępności prawdziwych ekranów dotykowych tablet może wydawać się niezbyt atrakcyjny, ale rysowanie na prawdziwym sprzęcie, a nie emulatorze, daje zupełnie inne odczucia. Polecam przekonać się o tym osobiście wszystkim fanon komputerów, niezależnie od posiadanej platformy sprzętowej.

> Opracował: Kamil Stokowski



Tak wyglądała kiedyś praca z piórem świetlnym podczas edycji tekstu oraz rysowania grafiki.











Rysowanie światłem

Wielu Czytelników na pewno traktuje tablety graficzne i pióra świetlne jako nietypowe urządzenia sterujące. Myślę, że tak się dzieje, bo w naszym kraju mało kto miał tego typu ekstrawaganckie dodatki do komputerów 8-bitowych. Trzeba powiedzieć, że mają one bardzo prosty sposób działania, a główne różnice wynikają z innego pomysłu na zewnętrzną formę.

Tablet graficzny działa podobnie do bardziej znanych "wiosełek", czyli paddle. Składają się one ze zmiennych rezystorów, czyli elementów, które mogą zmieniać ilość przepływającego przez nie prądu elektrycznego. Na przykład obracanie paddle w kierunku zgodnym z ruchem wskazówek zegara zwiększa przepływ prądu, natomiast obracanie go w kierunku przeciwnym do ruchu wskazówek zegara, zmniejsza przepływ, aż w końcu odcina całkowicie dopływ prądu.

Komputer, który obsługuje wiosełka, musi mieć zespół obwodów elektrycznych, który mogą zapewnić odczyt liczbowy wartości natężenia prądu, co jest proporcjonalne w stosunku do pozycji pokrętła paddle. Zazwyczaj jest to zero, gdy pokrętło jest obrócone całkowicie w jednym kierunku i 255, gdy jest całkowicie obrócone w przeciwnym kierunku. Bardzo podobnie działają joysticki do komputerów Apple, ale posiadają jeden rezystor na osi poziomej i drugim - w osi pionowej. W starszych tabletach graficznym zamiast zmiennych rezystorów stosowane są cienkie arkusze specjalnej folii. Kiedy naciskasz powierzchnię, prąd przepływa z wartością oporu zależną od miejsca naciśnięcia. Na przykład, jeśli folia jest ustawiona do rejestrowania w poziomie od lewej do prawej, naciśnięcie lewej krawędzi jest równoważne obracaniu paddle w celu uzyskania minimalnego oporu, natomiast naciśnięcie prawej krawędzi działa jak obracanie paddle dla uzyskania maksymalnego oporu.

Obszar roboczy tabletu składa się z dwóch arkuszy materiału, jednego przystosowanego do rejestrowania w poziomie i jednego do pomiarów pionowych. Ten sam zespół obwodów, który służy do odczytywania wartości wychylenia wiosełek lub joysticków można wykorzystać do odczytu danych z tabletu. Komputer interpretuje każdy z arkuszy jako paddle. Zwykle jeden odczyt daje poziome położenie punktu, a drugi odczyt - położenie pionowe.

Pióro świetlne działa na innej zasadzie. Aby to lepiej zrozumieć, najpierw powiedzmy, w jaki sposób tworzony jest obraz na ekranie CRT. Głównym elementem każdego kineskopowego urządzenia wyświetlającego - telewizora lub monitora - jest lampa, która umożliwia wyświetlanie obrazu za pomocą wiązki elektronów. Są one emitowane przez katodę, i trafiają – w zależności od rodzaju monitora – do trzech dział elektronowych (w przypadku monitorów kolorowych) bądź jednego działa elektronowego (w przypadku monitorów monochromatycznych). Działa formują elektrony z



STREFX XTXRI





... i Commodore 64.

katody w wąską wiązkę i "wystrzeliwują" ją. Wiązka przyśpieszana przez anodę uderza w powierzchnię ekranu pokrytą luminoforem wywołując jego świecenie. Aby dało się rozświetlić każdy punkt powierzchni ekranu wiązka musi być odchylana w dwóch kierunkach – pionowym i poziomym.

W najprostszej postaci pióro świetlne jest plastikowym cylindrem z fototranzystorem, włącznikiem światła. Kiedy pióro jest utrzymywane w pozycji "na ekranie", wiązka elektronów, które oświetlają ekran wyzwala fototranzystor powodując, że komputer odbiera sygoprogramowanie nał. Dalej musi sprawdzić, ile czasu upłynęło od rozpoczęcia działania pióra, a następnie obliczyć pozycję poziomą i pionową.

Obraz jest odrysowywany wiele razy na sekundę, dlatego pióro wykrywa strumień elektronów za każdym razem, gdy ekran zostanie odświeżony. Plamy światła są małe i za każdym razem pióro może powodować działanie w nieco innym punkcie, dlatego odczyty otrzymane z prostych urządzeń mogą być nieco niestabilne, zwłaszcza w przypadku poziomego położenia. Programy obsługujące takie urządzenia zazwyczaj wymagają naciśnięcia odpowiedniego klawisza na klawiaturze komputera, aby poinformować komputer, że odczytywanie informacji ma być wykonane w określonym momencie.

Bardziej zaawansowane, a przez to droższe pióra świetlne, mają dodatkowy zespół obwodów elektrycznych, który umożliwia utrzymanie ustalonych wartości, dzięki czemu odczyty nie zmieniają się za każdym razem, gdy zostanie narysowany ekran. Oznacza to, że odczyty będą znacznie stabilniejsze i że użytkownik nie musi używać dodatkowo klawiatury. Przełącznik w piórze informuje, kiedy należy zatrzymać bieżący odczyt. W niektórych wersjach przełącznik jest wbudowany w końcówkę pióra, dzięki czemu można ustalić dany odczyt, po prostu dociskając pióro do ekranu.

Jeśli chcesz używać pióra świetlnego lub tabletu graficznego we własnych programach, pamiętaj, że obsługę urządzenia należy napisać samodzielnie. Pióro nie będzie rysować po ekranie "samo z siebie", urządzenie to pozwoli Ci tylko odczytać liczby oznaczające pozycję na ekranie. Do Ciebie należy napisanie oprogramowania, które odszyfruje dane wejściowe z pióra lub tabletu, a następnie zrealizuje określoną funkcję w programie.

Wszystkim zainteresowanym działaniem pióra świetlnego polecam także zainteresować się nowszymi urządzeniami tego typu. Możemy kupić na przykład pióra wyposażone w diody LED. Więcej na ten temat można przeczytać pod poniższym adresem:

http://www.serwis-elektroniki.com.pl/wp-content/uploads/2017/03/170315_sr_1.pdf

Kamil Stokowski





Kocie nowości

Sylvester the Lumberjack

Właśnie powstaje nowa gra na Atari Lynxa! Jest to znana zręcznościówka gra o drwalu. Po raz pierwszy zobaczyłem ją na imprezie RetroKomp w Gdańsku, ale był to inny port. Z roku na rok była prezentowana na GameDev Compo na dwie platformy. Pierwsza wersja powstała na Amigę, natomiast druga, z tego co pamiętam, na Gameboya.

Gra jest prosta i polega na umiejętnym operowaniem drwalem, aby ciąć drzewo w odpowiednich miejscach, jednocześnie uważając aby nie zostać uderzonym gałęzią w głowę, co zawsze kończy się tragicznie dla naszego bohatera. Lumberjack zostanie wydany na cartidge'u w niewielkim nakładzie. Więcej informacji można znaleźć na poniższej stronie:

https://whitelynx.fi/

Bond w Shaken, not stirred

Podczas odbywającego się nieda-wno party eJag-fest w Niemczech zaprezentowano kompletną grę - Shaken, not



Świąteczna oprawa gry Fast Food 64.

stirred. Jak po tytule wnioskować można, produkcja oparta jest na filmach z Jamesem Bondem w roli głównej. W grze naszym zadaniem jest rozbroić bomby w różnych częściach miasta, w różnych loka-lizacjach takich jak magazyny, szkoły czy kasyno.



Lumberjack - gra znana z innych platform.

Aby wykonać zadanie musimy uzyskać przydatne informacje od pięknych kobiet, których Bond nie unika, a więc jest to dodatkowa atrakcja. Na miejsce, gdzie mamy rozbroić niebezpieczne ładunki dojeżdżamy autem. ale przeszkadzają nam w tym inni użytkownicy drogi. Po dojechaniu na miejsce często okazuje się, że pomieszczenia i budynki są chronione, dla-tego czeka nas strze-lanina, trochę w stylu słynnej gry Ope-ration Wolf.

Grę już zamówiłem i kompletną opinię będę mógł wydać później. Koszt wersji

> pudełkowej to 29,90 dolarów. Gra dostępna jest na tej samej stronie w sieci:

https://whitelynx.fi

Fast Food 64

Jaguarowa nowość na Święta Bożego Narodzenia! Objadamy się w grze Fast Food 64, w jej specjalnej świątecznej edycji.

Ta gra znana jest, między innymi, z tego, iż brała udział rok temu w GameDev Compo na imprezie RetroKomp. Teraz zyskuje - specjalnie na najbliższe Święta - nową grafikę oraz przyjemną muzykę. Dodane są nawet nowe efekty, czyli nasze "gęby", które się objadają.

Mamy też inne opcje, na przykład teleportację czy dopalacz rakietowy pozwalający osiągnąć na chwilę spore przyśpieszenie. Do tego dochodzą nowe ruchy naszego bohatera, a także "wrogów".

Gra na kilka godzin przed oddaniem naszego pisma do druku ukazała się dopiero w pierwszych wersjach, ale wygląda naprawdę sympatycznie i duch specjalnych gier na Boże Narodzenie może zostać zachowany. Produkcji szukajcie na stronie:

http://www.wavelgames.net

Piotr "Pitter" Krużycki



Bufor klawiatury

W tym artykule dowiesz się, jak wykorzystać niezwykle potężną funkcję w Atari - bufor klawiatury. Jest to zarezerwowany obszar pamięci komputera używany do tymczasowego zapisywania naciśnięć klawiszy, gdy klawiatura jest nieaktywna. Gdy klawiatura jest gotowa do wprowadzania znaków, wszystkie zapisane wcześniej klawisze zostaną wypisane na ekranie.

Dzięki buforowi znaki można wpisywać podczas działania programu, przewijania zawartości listingu, a nawet podczas zapisywania programu za pomocą instrukcji SAVE. Na początek spójrz na ten prosty program:

10 GOTO 10

Po uruchomieniu komputer zostanie "zablokowany" w nieskończonej pętli. Jeśli podczas działania tego programu wpiszesz jakieś znaki, Atari oczywiście zignoruje dane wejściowe. Po dodaniu bufora klawiatury co prawda nadal nic nie widać na ekranie, ale po zatrzymaniu programu za pomocą klawisza BREAK, wszystkie znaki, które wcześniej wpisałeś, zostaną wyświetlone.

Komputery takie jak IBM PC mają skomplikowane bufory klawiatury kon-

trolowane przez oddzielny mikroprocesor. Inne maszyny, takie jak Commodore 64 i VIC-20, mają proste dziesięcioznakowe bufory wbudowane w system operacyjny. Komputery Atari nie mają bufora, ale nasz program spowoduje utworzenie bufora o pojemności aż 100 znaków.

Program działa w następujący sposób: po każdym naciśnięciu klawisza zostanie sprawdzone, czy komputer jest aktualnie zajęty, czy też nie. Jeśli komputer nie jest przygotowany na przyjmowanie danych, liczba reprezentująca naciśnięty klawisz zostanie zapisana w buforze. Gdy tylko komputer będzie gotowy do odebrania danych, znaki zapisane w buforze zostaną wyświetlone na ekranie.

Listing z numerem 1 jest programem w Basicu, który ładuje kod języka maszynowego do pamięci. Może być to podprogram dla dowolnego innego programu w Basicu wymagającego wprowadzania danych z klawiatury. Po wpisaniu programu numer 1, zapisz go na dysku. Jeśli chcesz przetestować rezultaty swojej pracy, wpisz program numer 2 i uruchom za pomocą instrukcji RUN.

Jeśli pojawi się komunikat o błędzie lub komunikat SPRAWDZ LINIE DATA, oznacza to, że popełniłeś błąd podczas wpisywania programu numer 1. Należy wtedy dokładnie sprawdzić wszystkie instrukcje DATA. Gdy wszystko jest w porządku, komputer wyświetli napis:

WPISZ SWOJE IMIE:

i rozpocznie się odtwarzanie dźwięków. W trakcie, gdy gra muzyka, wpisz swoje imię i naciśnij RETURN. Po zakończeniu odtwarzania Twoje imię zostanie wypisane na ekranie.

Oczywiście to tylko jeden z przykładów zastosowania bufora klawiatury. Jeśli chcesz używać tej funkcji podczas wykonywania własnego programowania w Basicu, zmień linię 30150 na:

30150 END

a następnie uruchom program. Po wyświetleniu napisu READY wpisz: NEW. Bufor klawiatury będzie działał i możesz rozpocząć pisanie swojego programu.

Bufor może z pewnością poprawić funkcjonalność dowolnego programu wymagającego wpisywania danych przez użytkownika. Teraz możemy wpisywać znaki, nawet w sytuacji, gdy komputer wykonuje długą pętlę, co oszczędza czas. Ponadto program może być bardziej skomplikowany, a zaraz po odczytaniu danych z bufora klawiatury, możesz wywołać dodatkowe funkcje.

> Opracował: Marcin Libicki



```
Program nr 1
30000 REM BUFOR KLAWIATURY
30010 DATA 104, 173, 8, 2, 141, 96, 6, 173, 9, 2, 141, 97, 6,
           169, 0, 141, 14, 21, 2, 120, 169, 52, 141, 8, 2
30020 DATA 169, 6, 141, 9, 2, 169, 98, 141, 36, 2, 169, 6, 141,
           37, 2, 169, 192, 141, 14, 212, 169, 0, 133, 204
30030 DATA 133, 205, 88, 96, 173, 9, 210,n
         201, 159, 240, 36, 152.72, 173, 252,
         2, 201, 255, 240, 19, 164, 204, 192,
        100
30040 DATA 240, 9, 230, 204, 200, 173, 9,
         210, 153, 14 3, 6, 104, 168, 104, 64,
         165, 204, 197, 205.208, 231, 104, 1
        68, 76
30050 DATA 95, 6, 173, 252, 2, 201, 255, 2
         08, 35, 165, 204, 197, 205, 240, 23,
         230, 205, 164, 204, 192, 120, 176, 1
         5, 164
30060 DATA 205, 192, 120, 176, 9, 185, 14
         3, 6, 141, 252, 2, 76, 98, 228, 169, 0,
         133, 204, 133, 205, 76, 98, 228
30070 T = 0
30080 IF PEEK (521) = 6 THEN GOTO 3015
30090 FOR I = 1 TO 143
30100 READ N:T = T + N
30110 POKE 1535 + I, N
30120 NEXT I
30130 IF T <>18309 THEN PRINT "SPRAWDZ LINIE DATA":STOP JA
30140 A = USR(1536)
30150 RETURN
Program nr 2
10 REM KEYBOARD BUFFER TEST
20 GOSUB 30000
30 DIM NAME$ (30)
40 PRINT "WPISZ SWOJE IMIE:";
50 FOR I = 1 TO 500
60 A = RND(0) \times 255
70 SOUND 1, A, 10, 8
80 NEXT I
90 INPUT NAME $
100 PRINT "TWOJE IMIE TO: "; NAME$
110 END
```

43



WYKOPALISKA



FORT APOCALYPSE

Ostatnio, podczas jednej z retro imprez usiadłem do tei sprzed gry kilkudziesięciu lat i... okazało się, że się wcale nie zestarzała! Pamiętam, gdy grałem w nia w wieku około 10 lat. Co prawda dzisiaj moja wyobraźnia nie działa tak jak kiedyś, tzn. nie dobudowuję sobie scenariusza i całej otoczki, tylko po prostu gram, ale trzeba przyznać, że tak zwany "gameplay" jest nadal świetny i gra się wyśmienicie.

Gra powstała pierwotnie na 9-bitowe Atari, a została napisana przez Steve'a Halesa i wydana przez firmę Synapse Software znaną już wcześniej z tytułu takiego jak Protector. W dziale Wykopaliska opisywaliśmy dotąd gry dostępne tylko na platformę Atari, tutaj mamy pierwszy wyjątek, gdyż Fort Apocalypse został przeportowany także na Commodore 64. Szczerze mówiąc, nie wiem czy na C64 ta produkcja była popularna (była - przyp. Red.), ale na Atari była wielkim hitem. Tak ją przynajmniej pamiętam.

Z tego, co wiem, na Commodore 64 było więcej tego typu gier, więc może ktoś kto w nią nie grał kilkadziesiąt lat temu, a teraz zasiądzie za sterami śmigłowca i będzie penetrował podziemia. Tak - podziemia, gdyż akcja naszej gry rozgrywa się w 4 warstwach. Pierwszą jest powierzchnia ziemi, poniżej mamy dwie warstwy jaskiń, a najgłębiej znajduje się tytułowy fort. Fort Apocalypse nie jest typową, jak na ówczesne czasy, strzelaninπ. Wyróżnia ją fakt, iż jest bardziej skomplikowana, a nasz świat możemy dowolnie zwiedzać. Przykładowo nie jesteśmy zmuszeni lecieć tylko w prawą stronę jak w grze Scramble, mamy swobode wyboru. Fort Apocalipse przypomina mi grę Choplifter, z tym że jest od niej bardziej rozbudowany.

Nasz śmigłowiec wyposażony jest w działko i bomby. Mamy też bardzo przydatny radar, umieszczony na górze ekranu, który przekazuje sporo informacji o wrogach oraz ukształtowaniu terenu. Jak każde szanujące się urządzenie do przemieszczania, nasza maszyna

potrzebuje paliwa, Tutaj zasoby dość szybko się kończą i musimy lądować na specjalnie przygotowanych lądowiskach oraz ciągle uzupełniać cenny dla nas surowiec.

Mamy też inne lądowiska, gdzie zapisywany jest stan gry, czyli - po utracie życia startujemy właśnie z tego miejsca. W grze świetni są nasi wrogowie, do których można zaliczyć wiele obiektów - od dział rakietowych,



Grafika z okładki pudełkowego wydania.

przez działka, inne śmigłowce, aż do laserów. Musimy przebijać się naszym działkiem przez mury, rozbijać tarcze, uważać na teleporty lub umiejętnie z nich korzystać.

Cała rozgrywka jest bardzo szybka i nie pozwala się nudzić. Ciekawe są dalsze etapy i ogólnie im dalej, niżej się zagłębiamy, czujemy dreszczyk emocji. Naprawdę polecam tę grę nie tylko Atarowcom.

STREFA ATARI





Przebijanie się przez mury, ściany i skały to częste elementy w grze.

Poniżej: ekran ładowania i drugi poziom.



Zapomniałbym o ludziach, tak, tak mamy też misje ratunkową. Na każdym poziomie możemy odnaleźć i zabrać z sobą, podobnie jak w grze Choplifter, "swoich ludzi". Nie wiadomo tylko, czy są to cywile czy wojskowi, gdyż są jednokolorowo ubrani przez grafika. Ale to już nieistotny szczegół.

Po premierze gry Fort Apocalypse, firma Synapse Software istniała jeszcze 2 lat i wydała takie hity jak Blue Max, Pharaoh's Curse czy Shamus. Specjalizowała się w grach na Atari i jako pierwsze wszystkie produkcje były wydawane właśnie na tę platformę. W 1984 roku firma popadła w kłopoty finansowe za sprawą przejmowania rynku gier przez głównego konkurenta, czyli Commodore 64. Pod koniec 1984 roku Synapse zostało zakupione przez Brøderbund Software, czyli firmę dużo większą i została przez nią wchłonięta.

Z innych ciekawostek warto wiedzieć, że Fort Apocalypse sprzedał się w ilości około 75 tys. kopii. Myślę, że to dobry wynik, a jak na dzisiejsze czasy oczywiście niedościgniony.

Piotr "Piter" Krużycki



FORT APOCALYPSE

Większy dysk w systemie DOS 3.3

Ten nietypowy program zwiększa ilość miejsca na dyskach zapisanych w systemie DOS 3.3. Działa na każdym komputerze z serii Apple II z napędem dyskowym. Nie zajmuje wiele miejsca w pamięci i można go wpisać szybko w Basicu, tak jak za starych dobrych czasów.

Jeśli korzystasz z napędu dyskowego, wiesz, że przestrzeń dyskowa jest bardzo cenna. Jednym ze sposobów na zwiększenie pojemności dysku jest zakup specjalnego sprzętu, ale to kosztowna propozycja i związana z koniecznością "polowania" na portalach aukcyjnych.

Prezentowany przeze mnie program oferuje prostą alternatywę. Umożliwia formatowanie nowych dysków z maksymalnie pięcioma dodatkowymi ścieżkami, tworząc ponad 20 kilobajtów dodatkowego miejsca na dysku.

Wpisz listing prezentowany w ramce i zapisz go przez uruchomieniem. Program jest napisany w całości w Applesoft Basic. Najpierw zostanie wyświetlony monit o włożenie pustej dyskietki, która następnie zostanie zainicjowana. Ponieważ różne dyski pozwalają na inną liczbę dodatkowych ścieżek, program sformatuje tylko tyle dodatkowych ścieżek, ile będzie można używać bez obaw. Program automatycznie odczytuje i weryfikuje każdą dodatkową ścieżkę i jeśli ścieżka nie może być użyta, należy zrestartować formatowanie, używając kolejnej niższej wartości ścieżki. Po zakończeniu inicjalizacji program wyświetli liczbę poprawnie sformatowanych ścieżek.

Trzeba także wziąć pod uwagę, że różne dyski mogą nie być w stanie wykorzystać tej samej liczby dodatkowych ścieżek. Jeśli chcesz użyć zmodyfikowanego zapisu na innej dyskietce, dobrze jest wcześniej ustalić, czy możliwe jest uzyskanie dostępu do dodatkowych ścieżek.

Aby to zrobić, po prostu uruchom wpisany program na drugim nośniku i zanotuj liczbę ścieżek wyświetlanych po zakończeniu działania. Gdy znasz liczbę ścieżek, do których mogą mieć dostęp oba dyski, zmień liczbę w pierwszej linii programu i uruchom go ponownie.

Jak widać, korzystanie z dodatkowych ścieżek na dysku Apple jest zaskakująco łatwe. Program po prostu modyfikuje wartości używane przez procedury DOS-u, które inicjalizują dysk i tworzą jego katalog, czyli VTOC (Volume Table of Contents). Nośniki Apple są zwykle formatowane za pomocą 35 ścieżek.

Pierwsza instrukcja POKE w linii 130 wymusza na systemie DOS formatowanie większej ilości ścieżek poprzez zastąpienie wartości oznaczającej maksymalną ścieżkę. Pozostałe linie z POKE odpowiednio dostosowują VTOC i bitmapę. Mapa bitowa jest częścią VTOC, która pokazuje, gdzie znajdują się wolne sektory na dysku.

Każda ścieżka ma cztery bajty w mapie bitowej (dwa bajty nie są używane), a każdy bit reprezentuje odpowiedni sektor na ścieżce. Jeśli bit jest wyłączony (czyli ustawiony na wartość 0), oznacza to, że sektor jest już przydzielony. Jeśli bit jest włączony (czyli mamy wartość 1), sektor jest wolny.

Aby zapewnić prawidłową sytuację, w której VTOC i mapa bitowa uwzględniają dodatkowe ścieżki, ostatnie dwie instrukcje POKE w linii 130 ustawiają nowe wartości dla liczby ścieżek na dysku i rozmiaru bitmapy. Po sformatowaniu 40 ścieżek mapa bitowa ma 160 bajtów. Ważne jest, aby dysk mógł niezawodnie korzystać z dodatkowych ścieżek, dlatego linie 440-510 programu zawierają dane dotyczące procedury języka maszynowego, która sprawdza nowe ścieżki.

Odczytywany jest losowy sektor z każdej dodatkowej ścieżki, a następnie sprawdzane są błędy odczytu. Jeśli wystąpi błąd, zakładamy, że nie można uzyskać dostępu do ścieżki i inicjować dysku bez tej ścieżki.

> Opracował: Mariusz Wasilewski

Bajt	Błąd (oryginalne brzmieniej
00	Not used
01	Track of first catalog sector
02	Sector of first catalog sector
03	DOS release number (3.3, etc.)
04-05	Not used
06	Volume number
07-26	Not used
27	Max number of track/sector pairs
28-2F	Not used
30	Last allocated track
31	Direction of allocation
32-33	Not used
34	Number of tracks per disk
35	Number of sectors per track
36-37	Number of bytes per sector
38-3B	Bit map of track 0
3C-3F	Bit map of track 1
40-43	Bit map of track 2
BC-BF	Bit map of track 33
C0-C3	Bit map of track 34
C4-FF	Bit maps of additional tracks (if desired)

100 TRACKS = 40 : REM # OF TRACKS VALUE 110 BITMAPSZ = TRACKS * 4 : REM BIT MAP SIZE 120 FOR ML = 768 TO 829 : REM LOC OF ML ROUTINE 130 READ BYTE : POKE ML, BYTE: C HK5UM = CHK5UM + BYTE **140 NEXT** 150 IF CHKSUM < > 9911 THEN HOME : PRINT "ERROR IN DATA STATEMENTS" : GOTO 260 160 POKE 216, 0 : ONERR GOTO 270: REM RESET ONERR FLAG 170 POKE 48894, TRACK5 : POKE 46063, TRACK5 : POKE 44725, BITMAPSZ 180 HOME : PRINT "FORMATTING"; TRACKS; " TRACKS 190 PRINT CHR\$ (4); "INIT DISK BOOSTER, D1" 200 CALL 768: REM CAL VERIFY TRACKS ML ROUTINE AT \$0300 210 PRINT :"FORMATTING COMPLETE " ; TRACKS; " TRACKS WRITTEN" 220 PRINT 230 PRINT "DON'T FORGET TO LABEL" 240 PRINT "YOUR NEW DISK WITH" 250 PRINT "THE NUMBER OF TRACKS INITIALIZED!" 260 END 270 A = PEEK (222): REM ERROR CODE 280 ERRL = PEEK (218) + PEEK (219) * 256: REM ERROR LINE 290 IF A < > 8 THEN GOTO 350 300 HOME : PRINT "ERROR DETECTED 310 PRINT "RESTTING NEW TRACK VALUE" 320 PRINT "PRESS < RETURN > TO CONTINUE" 330 INPUT " "; IN\$: IF IN\$ < > " " THEN GOTO 330 340 TRACKS = TRACKS ! 1 : BITMA PSZ = TRACKS * 4 : GOTO 160 350 PRINT "ERROR NUMBER ";A; " DETECTED IN LINE "; ERRL 360 PRINT " CHECK DOS PROGRAMMERS MANUAL" 370 PRINT " FOR ERROR TYPE " 380 PRINT " AND CORRECT IT ACCORDINGLY" 390 GOTO 260 400 REM **(ERROR CODES FOR APPLE DOS 3.3** 410 REM ERROR #4 = WRITE PROTECTED DISK (REMOVE PROTECT TAB) 420 REM ERROR #11 = SYNTAX ERROR (CORRECT TYPOS) 430 REM 440 DATA 169, 0, 141, 235, 183, 141, 240, 183 450 DATA 169, 1, 141, 244, 183, 169, 16, 141 460 DATA 241, 183, 133, 209, 173, 254, 190 470 DATA 141, 236, 183, 56, 233, 35, 144, 22, 133 480 DATA 210, 169, 183, 160, 232, 32, 181 490 DATA 183, 176, 12, 206, 236, 183, 198 500 DATA 210, 208, 240, 198, 209, 208, 224

Menu rozwijane dla Apple II

W okresie rozwoju Macinsłynna firma Apple tosha spowodowała zmiany także w środowisku użytkowników 8bitowych maszyn. Wiele osób chciało mieć menu, ikony czy też pomoc na ekranie, podobnie jak łatwiejszy w użyciu graficzny interfejs użytkownika. Za pomoca mojego programu można dodawać atrakcyjne menu rozwijane, podobne do obecnego w systemie komputerów Macintosh. Menu można dodać do dowolnego programu w Basicu na Apple II z DOS-em 3.3 lub Pro-DOS-em.

Aby wszystko działało, potrzebne są dwa programy: podprogram w Basicu, który można łatwo dodać do końca dowolnego programu i procedura języka maszynowego, która tymczasowo zapisuje, a następnie przywraca tekst zawarty w menu rozwijanym. Basic pozwala przenieść cały ekran tekstowy w kilka sekund, natomiast asembler jest znacznie szybszy, co tutaj przydaje się znakomicie. Dodatkową zaletą jest fakt, że programista nie musi znać języka maszynowego, bo korzysta jedynie z rezultatów jego działania za pomocą instrukcji Basica.

Aby uruchomić menu rozwijane, musisz wpisać i zapisać oba programy

wymienione w ramce. Program numer 1 służy do tworzenia plików i automatyzapisuje procedure cznie języka maszynowego na dysk jako plik binarny o nazwie MOVE. Należy go wpisać i zapisz kopię, a następnie uruchomić. Pronumer 2 to natomiast aram przykładowy program w Basicu, który pokazuje rozwijane menu. Jak już wspomniałem, est przeznaczony do pracy z DOS-em 3.3 lub ProDOS-em. Jeśli używasz DOS 3.3, program trzeba wpisać dokładnie tak, jak widać na listingu. W przypadku ProDOS-u zmieniamy linię 150, tak jak poniżej:

75 150 HIMEN: 35840

Program ładuje plik MOVE z dysku, dlatego należy pamiętać, aby umieścić właściwy dysk w napędzie, zanim listing zostanie uruchomiony. Po uruchomieniu program symuluje prosty procesor tekstu z ekranem zawierającym tekst. Możemy pisać po ekranie i przesuwać kursor za pomocą klawiszy strzałek lub za pomocą kombinacji CTRL-J i CTRL-K (kursor w górę i w dół).

Po naciśnięciu klawisza ESC pojawi się menu rozwijane. Następnie możemy zmieniać wybór opcji menu za pomocą klawiszy kursora i zatwierdzić zmianę naciskając klawisz RETURN. Tekst znajdujący się "za menu" jest przywracany do poprzedniego stanu zaraz po wyjściu z menu, tak jak powinno to wyglądać na "starszym" sprzęcie. Ważną częścią programu demonstracyjnego jest podprogram zaczynający się od linii 63000. Procedura ta umożliwia dodawanie rozwijanych menu do własnych programów przy minimalnej ilości pracy. Generuje kształt okna i wywołuje MOVE w odpowiednim czasie. Wszystko, co musisz zrobić, to dodać linie 63000-63500 na końcu dowolnego programu w Basicu, a zgodnie nastepnie postępować Z poniższymi wskazówkami:

1. Twój program musi wykonać instrukcję BLOAD MOVE, tak jak w wierszach 180-190, czyli przed wywołaniem procedury języka maszynowego.

2. HIMEM należy ustać na początku (patrz linia 150), zanim zostaną zadeklarowane zmienne lub otwarte pliki. Dla DOS 3.3 używamy wartości 36914, natomiast dla ProDOS-u -35840.

3. Zmienną NN należy ustawić na liczbę równą maksymalnej liczbie pozycji, które mają być dostępne w największym menu (patrz linia 160). Podprogram menu automatycznie określa liczbę pozycji w każdym menu i odpowiednio dostosowuje rozmiar okna.

3. Zmienna MM\$ musi być tablicą (patrz linia 170) i odpowiada ona liczbie potrzebnych nazw opcji możliwych do wyboru. Każdy element tablicy wypełni-

STREF# #PPLE

amy ciągiem zawierającym nazwę. Należy to zrobić przez odczytanie ciągów z linii DATA (tak jak w liniach 200-220) lub definiując każdy ciąg za pomocą instrukcji takich jak:

MM\$(1)="Opcja menu"

4. Zmienna TITLE\$ musi być zdefiniowana jako tytuł menu (patrz linia 470). Podprogram menu automatycznie centruje tytuł na ekranie.

5. Linie 690-850 pokazują, jak używać MOVE do dodawania instrukcji do programów bez utraty oryginalnego ekranu. W tym przypadku kombinacja CTRL-I służy do wybierania instrukcji.

Jeśli mamy mysz zgodną z Apple, możemy jej użyć do wywołania menu i wskazywania opcji. Wymaga to kilku zmian w programie. Najpierw usuwamy linie 320, 330 i 63360-63460. Następnie zmieniamy linie 310, 450 i 63350 w następujący sposób:

E9 310 PRINT "PRESS ESC KEY OR M Ouse Button for Menu"; 98 450 Goto 311 66 63350 HTAB 3: VTAB select + 2 : Inverse : Print ">" C HR\$ (8):s Normal

Dalej dodajemy poniższe linie:

- D2 235 PRINT : HOME : PRINT D\$"P R442"s PRINT CHR5 (1): PRI NT D\$"PR#0"
- 0A 311 VTAB 15: HTAB 1: PRINT CH R\$ (13)D\$"IN#2"
- 55 312 VTAB 23: HTAB 405 INPUT " ";X,Y,B0
- IF 313 IF B0 = 1 OR B0 < 0 THEN 316
- 80 314 VTAB CV: HTAB CH: FLASH : Print " ";: Normal
- AB 315 GOTO 312
- C2 316 PRINT D\$"IN#0"
- 57 317 IF B0 = 1 THEN IN\$ = CHR5 (27): GOTO 319
- 8F 318 IN\$ = CHR\$ (PEEK (163 84) - 128)

```
34 319 POKE - 16368,0
BD 320 VTAB CV: HTAB CH: PRINT "
        ۳;
F7 395 IF CH > 0 THEN HTAB CH
79 396 IF CV > 0 THEN VTAB CV
F8 63360 VTAB 1: HTAB LMAX + 5:
       PRINT : HTAB LMAX + 5:
       PRINT D$"IN#2": VTAB 1:
        HTAB LMAX + 5: INPUT "
       ";X0,Y0,B0
61 63370 IF B0 = 1 THEN 63430
80 63380 Y0 = INT (Y0 / 10)
6B 63390 VTAB SELECT + 2: HTAB 3
       : PRINT " ";
C9 63400 SELECT = Y0: IF SELECT
       > NITEMS THEN SELECT =
       NITEMS
3D 63410 IF SELECT < 1 THEN SELE
       CT = 1
04 63420 GOTO 63350
DC 63430 PRINT D$"IN#0"
```

Jeśli nasz komputer korzysta z Pro-DOS-u, zmieniamy linię 311 w następujący sposób:

01 311 VTAB 15: HTAB 1: PRINT D\$ "IN#2"

PR#2 i IN#2 w liniach 235, 311 oraz 63360 zakładają, że interfejs myszy znajduje się w gnieździe numer 2. Jeśli interfejs znajduje się w innym gnieździe, odpowiedni numer należy wpisać w tych liniach. Jeśli korzystamy z Apple IIc, trzeba użyć PR#4 i IN#4, zamiast PR#2 i IN#2, bowiem mysz znajduje się w logicznym gnieździe numer 4.

Po wprowadzeniu tych zmian mysz powinna działać. Oczywiście zamiast klawisza RETURN można używać także przycisku myszy do wybierania opcji.

Program numer 1

B7 100 REM BASIC PROGRAM FOR AD 110 REM GENERATING THE 44 120 REM BINARY FILE 29 130 REM 'MOVE' 4C 140 HOME

IC 150 VTAB 12: PRINT "WORKING . .." 92 160 FOR I = 0 TO 459 21 170 READ A CC 180 POKE 36915 + I,A D1 190 VTAB 12: HTAB 13: PRINT I + 1 DE 200 NEXT I FF 210 PRINT CHR\$ (4)"BSAVE MOVE ,A36915, L460" 2A 220 PRINT : PRINT "DONE!" AB 230 DATA 173,89,170,72,165,21 7,72,165,118,72,169 3F 240 DATA 2,133,118,169,255,13 3,217,169,191,133,51 5D 250 DATA 169,0,133,243,76,86, 144,76,86,76,86 D8 260 DATA 76,86,169,80,133,133 ,169,144,160,0,162 33 270 DATA 5,32,254,144,76,104, 144,76,104,169,102 35 280 DATA 133,133,169,144,160, 0,162,1,32,254,144 97 290 DATA 169,0,141,80,144,169 ,4,141,81,144,173 F5 300 DATA 81,144,201,8,48,14,2 08,9,173,80,144 60 310 DATA 201,0,144,5,240,3,76 ,234,144,173,80 7D 320 DATA 144,141,161,144,173, 81,144,141,162,144,173 99 330 DATA 0,16,141,82,144,169, 0,141,83,144,24 D1 340 DATA 169,255,109,102,144, 141,84,144,169,145,109 0D 350 DATA 103,144,141,85,144,1 73,84,144,141,204,144 8D 360 DATA 173,85,144,141,205,1 44,173,82,144,141,0 D9 370 DATA 16,24,173,102,144,10 5,1,141,102,144,173 6E 380 DATA 103,144,105,0,141,10 3,144,238,80,144,208 63 390 DATA 3,238,81,144,76,127, 144,104,133,118,104 35 400 DATA 133,217,104,141,89,1 70,169,141,141,1,2 C2 410 DATA 169,1,133,52,96,133, 134.132.135.160.0 88 420 DATA 169,0,145,133,200,20 8,2,230,134,138,208 F3 430 DATA 4,198,135,48,4,202,7 6,4,145,96,173

STREFA APPLE

7 L.

83	440	DATA 89,170,72,165,217,72
		,165,118,72,169,2
BA	450	DATA 133,118,169,255,133,
		217,169,191,133,51,169
44	460	DATA 0,133,243,76,60,145,
		76,60,76,60,76
E6	470	DATA 60,169,54,133,133,16
		9,145,160,0,162,5
84	480	DATA 32,228,145,76,78,145
		,76,78,169,76,133
7A	490	DATA 133,169,145,160,0,16
		2,1,32,228,145,169
E7	500	DATA 255,141,54,145,169,1
		45,141,55,145,173,55
5A	510	DATA 145,201,149,48,14,20
		8,9,173,54,145,201
85	520	DATA 255,144,5,240,3,76,2
		08,145,173,54,145
A7	530	DATA 141,135,145,173,55,1
		45,141,136,145,173,0
1E	540	DATA 16,141,56,145,169,0,
		141,57,145,24,169
BØ	550	DATA 0,109,76,145,141,58,
		145,169,4,109,77
74	560	DATA 145,141,59,145,173,5
		8,145,141,178,145,173
4D	570	DATA 59,145,141,179,145,1
		73,56,145,141,0,16
0E	580	DATA 24,173,76,145,105,1,
		141,76,145,173,77
04	590	DATA 145,105,0,141,77,145
		,238,54,145,208,3
07	600	DATA 238,55,145,76,101,14
		5,104,133,118,104,133
22	610	DATA 217,104,141,89,170,1
		69,141,141,1,2,169
33	620	DATA 1,133,52,96,133,134,
		132,135,160,0,169
41	630	DATA 0,145,133,200,208,2,
		230,134,138,208,4
C 9	640	DATA 198,135,48,4,202,76,
		234,145,96
۳10	oyral	
10	100	REM LINES 150-850 ADE
DB	110	REM & SAMPLE DDACDAM
F۵	120	REM DEMONSTRATING
D6	130	REM PILLI-DOWN MENUS
RA	140	REM
5B	150	HIMEM: 36914: REM FOR DOS
		3.3 ONLY. FOR PRODOS USF
		35840

14 160 NN = 20: REM MAXIMUM NU MBER OF ITEMS IN ANY MENU C6 170 DIM MM\$(NN): REM MM\$=MENU SELECTIONS 62 180 D\$ = CHR5 (4) 5C 190 PRINT D\$"BLOAD MOVE" FD 200 FOR I = 1 TO 5 BE 210 READ MMS(I) E2 220 NEXT I 4B 230 HOME 41 240 FOR I = 1 TO 15 57 250 PRINT "THIS IS A SAMPLE P ULL-DOWN MENU. ": EA 260 NEXT I B3 270 CV = 13:CH = 16 12 280 VTAB 21: HTAB 1: PRINT "------------": REM 39 DA SHES BE 290 PRINT TAB(5)"USE ARROW K EYS TO MOVE CURSOR" 35 300 PRINT TAB(5)"PRESS CTRL-I FOR INSTRUCTIONS" 36 310 PRINT TAB(8)"PRESS ESC K EY FOR MENU "; 8A 320 VTAB CV: HTAB CH 91 330 GET IN\$ 62 340 IF IN\$ = CHR\$ (9) THEN GO SUB 690 31 350 IF IN\$ = CHR\$ (27) THEN 4 70 C4 360 IF IN\$ = CHR\$ (8) THEN CH = CH - 1 71 370 IF IN\$ = CHR\$ (21) THEN C H = CH + 1D9 380 IF IN\$ = CHR\$ (11) THEN C U = CU - 153 390 IF IN\$ = CHR\$ (10) THEN C V = CV + 1BA 400 IF IN\$ > CHR\$ (31) THEN P RINT IN\$;:CH = CH + 1: IFCH > 40 THEN CH = 1:CV = CV + 1 73 410 IF CH < 1 THEN CH = 1 C6 420 IF CH > 40 THEN CH = 40 7E 430 IF CV < 1 THEN CV = 1 4F 440 IF CV > 20 THEN CV = 20 9A 450 GOTO 320 36 460 REM THE FOLLOWING LINE AC TIVATES THE MENU 33 470 TITLE\$ = "MENU": GOSUB 63 040 59 480 ON SELECT GOTO 280,490,50

0,510,590 54 90 HOME : PRINT "THE FIRST F UNCTION OF YOUR PROGRAM G OES HERE": GOTO 520 74 500 HOME : PRINT "THE SECOND FUNCTION OF YOUR PROGRAM GOESHERE": GOTO 520 08 510 HOME : PRINT "THE THIRD F UNCTION OF YOUR PROGRAM G OES HERE": GOTO 520 3F 520 VTAB 24: PRINT "PRESS ANY KEY TO CONTINUE ... "; D7 530 GET 65 35 540 FOR I = 1 TO NITEMS 97 550 MM\$ (I) = "" ED 560 NEXT I **DØ 570 RESTORE** 1F 580 GOTO 200 25 590 HOME : PRINT "GOOD-BYE!": END 99 600 DATA "LEAVE MENU" CD 610 DATA "FIRST SELECTION" 71 620 DATA "SECOND SELECTION" **B0 630 DATA "THIRD SELECTION"** AE 640 DATA "QUIT PROGRAM" 98 650 END 93 660 REM **8E 670 REM INSTRUCTIONS** 97 680 REM 6C 690 CALL 36915 D5 700 HOME : INVERSE : PRINT BL **ANKS** DC 710 VTAB 1: HTAB 14: PRINT "I **NSTRUCTIONS": NORMAL : VT AR 3** 68 750 PRINT "FOR THIS SAMPLE PR OGRAM, YOU CAN MOVE" 12 760 PRINT "THE CURSOR WITH TH E ARROW KEYS AND TYPE" 39 770 PRINT "ON THE SCREEN. WH EN YOU PRESS ESC, THE" 64 780 PRINT "COMPUTER WILL DISP LAY A PULL DOWN MENU." **30 790 PRINT "USE THE ARROW KEYS** TO MOVE THE SELEC-" 47 800 PRINT "TION CURSOR TO THE DESIRED OPTION, THEN" 9A 810 PRINT "PRESS RETURN TO SE LECT IT." 42 820 VTAB 24: PRINT "PRESS ANY KEY TO CONTINUE ... "; DA 830 GET A\$

STREF# #PPLE

61 840 CALL 37145 22 850 RETURN A5 62999 REM #63000 24 63000 REM EA 63010 REM PULL-DOWN MENU 81 63020 REM SUBROUTINE 3C 63030 REM 90 63040 BLANK\$ = " ": REM 39 SPACES A4 63050 LMAX = 0:NITEMS = 0 53 63060 REM DETERMINE MENU SIZE 24 63070 FOR II = 1 TO NN 59 63080 IF MM\$(II) = "" THEN 63 120 A3 63090 LL = LEN (MM\$(II)) 62 63100 IF LL > LMAX THEN LMAX = LL C2 63110 NITEMS = NITEMS + 1 CC 63120 NEXT II 65 63130 IF LMAX > 28 THEN PRINT "NAME IS TOO LONG": EN D

83 63140 REM SAVE SCREEN TEXT 98 63150 CALL 36915 A3 63160 REM DISPLAY MENU BA 63170 POKE 32,5: POKE 33,LMAX + 5: POKE 34,0: POKE 3 5,NITEMS + 4: REM SET T

EXT WINDOW FOR MENU SIZ Ε 6F 63180 HOME 32 63190 INVERSE : PRINT LEFT\$ (BL\$, LMAX + 5)D4 63200 VTAB 1: HTAB 3 + ((LMAX - LEN (TITLE\$)) / 2); PRINT TITLE\$ C0 63210 FOR II = 1 TO NITEMS + 2 0C 63220 VTAB II + 1: HTAB 1: PR INT " "; 6C 63230 HTAB LMAX + 5: PRINT " Ч÷. E0 63240 NEXT II C0 63250 POKE 35,24 0F 63260 PRINT LEFT\$ (BL\$,LMAX + 5); 17 63270 POKE 35.NITEM5 + 4 D0 63280 VTAB 1 7D 63290 NORMAL AB 63300 FOR II = 1 TO NITEMS 78 63310 HTAB 4: VTAB II + 2: PR INT MM\$(II) D4 63320 NEXT II 03 63330 REM MAKE SELECTION 99 63340 SELECT = 1

: PRINT ">" CHR\$ (8): 43 63360 GET SELECT\$ 91 63370 HTAB 3: VTAB SELECT + 2 : PRINT " " A4 63380 IF SELECT\$ = CHR\$ (13) THEN 63480 DF 63390 IF SELECT\$ < > CHR\$ (10) AND SELECT\$ < > CHR\$ (21) THEN 63430 18 63400 SELECT = SELECT + 1 06 63410 IF SELECT > NITEMS THEN SELECT = 1 04 63420 GOTO 63350 56 63430 IF SELECT\$ < > CHR\$ (11) AND SELECT\$ < > CHR\$ (8) GOTO 63350 40 63440 SELECT = SELECT - 1 83 63450 IF SELECT < 1 THEN SELE CT = NITEMS 24 63460 GOTO 63350 5C 63470 REM RESTORE SCREEN TEXT B3 63480 CALL 37145 94 63490 POKE 32,0: POKE 33,40: POKE 34,0: POKE 35,24: REM RETURN THE TEXT WI NDOW TO NORMAL 72 63500 RETURN





88 63350 HTAB 3: VTAB SELECT + 2

AMIGA.net.pl



Nowe gry dla komputerów retro Kolejne produkcje w przygotowaniu!

Macintosh kontra MS-DOS _{Część 2}.

W tym odcinku kontynuujemy dość nietypowe porównanie komputerów Apple z pecetem. Jak zauważyliście, w zasadzie analizuję architekturę sprzętową i osiągi poszczególnych komponentów, a nie funkcje samego oprogramowania. Takie zestawienie jest moim zdaniem najlepszą okazją, aby pokazać różnice, bez wchodzenia w bezowocne dyskusje na temat wyższości konkretnych programów lub różnych wersji systemu operacyjnego.

Komputery Apple jest łatwiejsze do porównania, ponieważ wszystkie były produkowane przez jedną firmę. Mac Plus i SE wykorzystują procesory 68000 pracujące z częstotliwością ok. 8 MHz. Ich porty SCSI umożliwiają transmisję odpowiednio na poziomie 350 i 700 kilobajtów na sekundę. Komputer Mac II korzysta z procesora 68020 z koprocesorem zmiennoprzecinkowym 68881 (FPU), działającym z częstotliwością około 15,8 MHz.

Port SCSI działa z maksymalną szybkością transferu danych około 1,2 megabajta na sekundę. Dostęp do kart NuBus zajmuje około 800 do 1000 ns. Mac IIx, IIcx i SE/30 używają procesora 68030 z 68882 FPU. Komputer Mac IIci pracuje z procesorem 68030 z częstotliwością 25 MHz, z obsługą 80-bitowych pamięci DRAM, gniazdem dla zewnętrznej karty pamięci podręcznej, obsługą trybu "burst-fill" i nieco szybszymi czasami cyklu NuBus.

Aby uzyskać maksymalną wydajność, Macintoshe muszą zostać przyspieszone za pomocą akceleratorów, podobnie jak przykładowo nowsze modele Amigi. Jedną z takich kart jest akcelerator Daystar z procesorem 68030 z zegarem 50 MHz. Zastępuje on fabryczny procesor 68020 w Macintoshu II i dodaje 32 kilobajty pamięci podręcznej. Ponadto szybka magistrala może łączyć akcelerator Daystar z kartami graficznymi, SCSI i kartami pamięci.

W przypadku komputerów PC, system MS-DOS korzysta z procesorów 8086, 286 lub 386, 386SX oraz 486. Jeśli porównany maszyny dostępne w tym samym okresie, co wspomniane Macintoshe, szybkości procesorów mieszczą



Komputer Macintosh SE w działaniu.

STREF# #PPLE



Komputer IBM PC 5710 z procesorem 6 MHz.

się w zakresie od 8 MHz do 33 MHz, a wbudowana pamięć podręczna nie ma 32 kilobajtów i co ważniejsze, nie ma możliwości rozszerzenia pamięci podręcznej.

Szybkości magistrali ISA wynoszą od 1,2 do 8 milionów bajtów na sekunde, chociaż ta ostatnia wartość wydaje się dotyczyć wyłącznie operacji DMA. Systemy EISA mają dwa razy większą przepustowość, przede wszystkim, ponownie, w stosundku do DMA. Systemy MCA (ang. Micro Channel Archiwysokie prędkości tecture) maja transmisji danych, bliskie najlepszej wydajności EISA, ale ulegają jeszcze szybszemu rozproszeniu w operacjach dostępu losowego.

Wszystkie systemy obsługują DMA i mogą używać kontrolerów dysków, których wydajność jest głównie ograniczona przez szybkość transmisji danych głowicy dysku. Buforowanie kontrolerów dysków w niektórych modelach znacznie poprawia wydajność dysku.

Ogólnie rzecz biorąc, systemy MS-DOS były zaprojektowane tak, aby

ograniczały procesor, a nie ograniczały pamięci, podobnie jak ówczesne komputery Macintosh. Maszyna z procesorem 386 przewyższa ok. czterokrotnie komputer Mac Plus, w niektórych zadaniach nie-zmiennoprzecinkowych przy użyciu porównywalnych komercyjnych programów. Podstawowy IBM AT (z zegarem 8 MHz) potrafi być nawet ponad dziesięć razy szybszy niż Plus w testach graficznych.

Oczywiście wszystko zależy od zastosowanej karty graficznej, bowiem jeśli komputer ma służyć do korzystania z programów takich jak AutoCAD, trzeba w nim zamontować szybką kartę, która kiedyś powodowała, że cały zestaw był dużo droższy niż Macintosh. Tak więc dyskusję należy ograniczyć najlepiej do konkretnego zestawy PC. Sprzęt z wyższej półki spowoduje, że ze względu na ówczesną cenę, komputer należy porównywać z Macintoshem z procesorem 68030.

Rodzina 68k zapewnia dużą liczbę trybów adresowania, które użyte w połączeniu z dużym zestawem rejestrów, pozwalają na użycie złożonych struktur danych bez zmniejszania wydajności. Z tego powodu zestaw narzędzi dostępnych dla Macintosh od początku został zaprojektowany zgodnie z zasadami programowania obiektowego. Najlepszym tego przykładem jest Dialog Manager, czyli dwie podklasy Window Managera (okna dialogowe i powiadomienia), który z kolei jest podklasą QuickDraw.

Na poziomie systemu operacyjnego hierarchiczny system plików (czyli HFS) systemu Macintosha wyraźnie przewyższa systemy MS-DOS i OS/2, nawet tak zwany system plików o wysokiej wydajności. Ta wyższość jest widoczna zarówno pod względem funkcjonalności, jak i wydajności.

HFS jest po prostu szybszy, rzadko wymaga więcej niż dwóch operacji dostępu do dysku, aby odczytać lub zapisać blok pliku. Podkatalogi były niejawne już w oryginalnym systemie plików Macintosh (MFS). Oznaczało to, że użytkownicy (i programiści) nie musieli uczyć się skomplikowanych ścieżek dostępu.

Z tego, co ustaliłem Apple około roku 1990 wydawał rocznie około 400 milionów dolarów na badania i rozwój. Jest to wystarczająco duża ilość, aby zaprojektować wdrożyć optymalne rozwiązania. Niestety wydajność dowolnego systemu może zepsuć nieprawidłowy projekt aplikacji, ale w przypadku Apple programy zawsze były wysokiej jakości. Jeśli ktoś chce zobaczyć przykłady dobrego oprogramowania z tamtych lat, proponuję aplikacje takie jak WingZ, WriteNow czy Think C.

W świecie MS-DOS, Microsoft wyznacza standardy, a programiści pracują wokół nich. Pomijanie funkcji DOS-u, a nawet BIOS-u, jest bardzo częstym sposobem osiągnięcia większej szybkości pracy. Należy pamiętać, że MS-

PHILIPS	P200	8808 / 200							
Microsof Copyrigh	t MS-DC t 1981	IS version ,82,83 Mic	2.11 rosoft	Corp.					
Command Current Enter ne Current Enter ne	v. 2.1 date is w date time is w time	1 5 Tue 27.1 28.07.20 5 8.32.27 10.32.20	1.1984 08 .84						
Volume Directo	in driv ry of	ve A is SY A:Ö	STEM-U	1-0					
COMMAND SYS MORE MSSORT CONFIG 2	COM COM COM EXE SYS 4 File	REBODT RECOVER EDL IN PCSORT INSTPHI (s) 649	CDM CDM CDM EXE BAT 215 by	CLR FC EXE2BIN ANSI INSTIBM tes free	CDM EXE EXE SYS SYS	DISKCOPY FIND FORMAT INSTIBM INSTPHI	COM EXE COM BAT SYS	CHKDSK DEBUG PRINT LINK	COM COM COM EXE
A>									

System MS-DOS drugiej generacji na komputerze Philips P2000c.

DOS rozpoczął swoje życie jako QD-DOS w systemie operacyjnym CP/M dla 8086. W rzeczywistości oznacza to, że MS-DOS początkowo działał tylko jako zbiór procedur i struktur danych. Nie był to więc kompletny zintegrowany system. Firma Microsoft musiała długo i ciężko pracować, aby usunąć ograniczenia MS-DOS.

W trakcie tego procesu wiele aplikacji straciło kompatybilność i wymagały one zmian, aby mogły działać w systemie 3.3. Jednak program pod kontrolą MS-DOS może działać szybciej niż porównywalna aplikacja dla Macintosha, szczególnie jeśli za kryterium przyjmiemy operacje zmiennoprzecinkowe.

Komputer firmy Apple musi wykonać dużo większą ilość operacji ze względu na złożony interfejs graficzny. Prosty tryb tekstpwy jest szybszy, ale z pewnością nie jest wygodniejszy, dlatego Microsoft w ramach konkurencji szybko zapowiedział własny system okienkowy, a jego niską wydajność w stosunku do środowiska GEM (lub pochodnych) jest dzisiaj znaną ogólnie cechą.

Podsumowując, bez uciekania się do niepewnych benchmarków, widzimy, że architektura Macintosha miała przewagę systemem MS-DOS na wszystkich poziomach, od podstawowego sprzętu poprzez oprogramowanie systemowe i dodatkowe aplikacje. Nie mówię tego wyłącznie jako entuzjasta sprzętu Apple, ale sprawa była szeroko omawiana w prasie branżowej.

Z drugiej strony, wiele zagranicznych publikacji z przełomu lat '80-tych i '90tych faworyzowało maszyny z MS-DOS-em z różnych powodów, natomiast w artykułach związanych z Macintoshem pojawiały się głosy, że wydajność serii procesorów 680x0 nie jest wystarczająca i zapowiadano kolejne generacje mające walczyć z rynek PC.

Historia dla naszej platformy potoczyła się bardzo szczęśliwie, ale trzeba też zwrócić uwagę, że pozostałe maszyny oparte o procesory Motoroli odeszły w przeszłość.

> Opracował: Mariusz Wasilewski



Globalne zapisywanie zmiennych

Modyfikowanie linii w programach w Basicu może być czasochłonne w przypadku danych zapisanych utraty wewnatrz zmiennych. Dlatego pokażę program w języku maszynowym, który rozwiązuje ten problem. Zapisuje on automatycznie resetuje i. wskaźniki związane ze zmiennymi, umożliwiając latwe przerywanie programów w celu modyfikacji i wyszukiwania błędów.

W Applesoft Basic tracisz zmienne za każdym razem, gdy program jest modyfikowany. Jest to szczególnie kłopotliwe podczas częstych zmian, na przykład, gdy musisz wyeliminować błąd logiczny. Za każdym razem, gdy dana linia jest zmieniana, konieczne jest ponowne wykonanie całego programu z powodu utraty wartości zmiennych. Gdy program wymaga długich obliczeń, wymóg ten powoduje, że modyfikacja programu jest powolnym i bardzo frustrującym procesem.

Używając tego krótkiego programu w języku maszynowym, możesz uniknąć wielu kłopotów. Uruchomiony program można przerwać za pomocą kombinacji klawiszy CTRL-C. Poszczególne wiersze programu można zmieniać, dodawać je lub usuwać, a wykonanie można wznowić za pomocą polecenia CONT lub GOTO. Wszystkie zmienne będą cały czas dostępne, gotowe do użycia, pod warunkiem, że nie wprowadzono żadnych nietypowych poleceń, które mogłyby zakłócić przechowywane zmienne lub ich wskaźniki.

Przedstawiona modyfikacja jest realizowana poprzez wywołanie nowej procedury kodu maszynowego, zamiast zwykłej procedury KEYIN, gdy wymagane jest użycie klawiatury. Całość składa się z dwóch części: funkcjonalnej części programu i krótkiej sekwencji inicjalizacji, która musi zostać uruchomiona w celu połączenia z Basiciem. Procedura inicjalizacyjna ustawia wskaźnik KSW na adresy \$38-\$39, aby wywołać funkcję.

Dodawana jest też nowa procedura do sekwencji RESET, aby podprogram pozostawał aktywny nawet po zresetowaniu systemu. Stan klawiatury jest odczytywany, a następnie program zalub pisuje odczytuje określone wskaźniki informujące Applesoft, gdzie znajdują się zmienne w pamięci. Procedura zajmuje 96 bajtów, w tym siedmiobajtowy obszar RSAVE i może być umieszczony w dowolnym miejscu w pamięci. Program numer 1 można wprowadzić najszybciej z poziomu monitora.

Aby śledzić zmienne, Basic ma cztery wskaźniki na zerowej stronie, które definiują początek tablicy dla zmiennych, koniec pamięci dla zmiennych i początek pamięci przyporządkowanej dla ciągów tekstowych. Pierwszy wskaźnik jest automatycznie ustawiany na końcu programu przez Applesoft, gdy program jest załadowany, poprzez wprowadzenie lub usunięcie linii programu. Wskaźnik ten można zmienić na wyższą wartość za pomocą instrukcji LOMEM lub POKE, umożliwiając programistom pozostawienie odstępu w pamięci pomiędzy programem a tabelami zmiennych. Inne wskaźniki nie są bezpośrednio dostępne.

Na początku programu, ostatni wskaźnik jest ustawiony na HIMEM. Ponieważ zmienne są przypisywane przez program, wskaźniki są aktualizowane. Zmienne nigdy nie sa usuwane z tabel zmiennych, natomiast wskaźniki nigdy nie zmniejszają wartości podczas wykonywania programu. Po wprowadzeniu nowej linii wskaźniki są resetowane do wartości domyślnych, dzięki czemu z poziomu Basica wydaje się, że nie zdefiniowano żadnych zmiennych. Jednak pozostają one cały czas gotowe do użycia.

Jeśli wskaźniki zostaną zapisane przed ich zresetowaniem, a następnie będą przywrócone po wprowadzeniu nowej linii, Applesoft może ich użyć. Można to zrobić za pomocą polecenia monitora M (Move), aby przechowywać wskaźniki w wygodnej lokalizacji, powrócić do Basica, aby dokonać zmian programu, a następnie ponownie użyć monitora, aby przywrócić zmienne WΖ obszaru przed kontynuowaniem pamięci wykonywania listingu.

Nasza procedura wykonuje te operacje automatycznie za pomocą funkcji SAVE (zapisywanie wskaźników) i RESTORE (przywracanie wskaźników) za każdym razem, gdy odczytywana jest klawiatura.

STREFA APPLE



Odpowiednią operację wybiera się, porównując bieżącą wartość wskaźnika z końcem pamięci zmiennej (\$6D-\$6E) z zapisaną wartością tego wskaźnika. Jeśli zapisana wartość jest mniejsza niż bieżąca wartość wskaźnika, wykonywana jest operacja SAVE, natomiast operacja RESTORE występuje, jeśli bieżąca wartość wskaźnika jest mniejsza niż wartość zapisana.

Zwykle oznacza to, że po zaktualizowaniu wskaźników są one zapisywane przy następnym użyciu klawiatury. RE-STORE występuje po wprowadzeniu z klawiatury tylko wtedy, gdy wskaźniki zmiennych zostały zresetowane do wartości domyślnych.

Program dokonuje tego porównania za każdym razem, gdy odczytuje stan klawiatury, chyba że wprowadzono kombinację CTRL-C. W takim przypadku operacja SAVE jest wykonywana niezależnie od bieżącej wartości wskaźników. Ten wyjątek jest konieczny, aby umożliwić wyczyszczenie tabel zmiennych. Aby wyczyścić zmienne, należy wprowadzić polecenie CLEAR, a następnie nacisnąć klawisze CTRL-C.

Aby nauczyć się korzystać z nowej procedury najlepiej uruchomić Program numer 2. Jeśli procedura będzie już działać, wykonanie programy zatrzyma się na linii 110 wyświetlając błąd składniowy (SYNTAX ERROR) z powodu błędnie napisanego słowa NEXT. W tym momencie obszar przechowywania zawiera domyślne wartości wskaźników podczas zmiennych, gdy same wskaźniki zawierają aktualne wartości przypisane przez Applesoft. Wartości te muszą zostać zapisane przed zmianą linii 110. Wpisanie poniższego wiersza:

110 NEXT I

lub naciśnięcie dowolnego klawisza) spowoduje następujące działanie: gdy wprowadzony zostanie pierwszy znak linii, wykonywana będzie operacja SAVE, zachowująca wskaźniki zmiennych. Kiedy naciśniesz RETURN, Applesoft wykona linię, sprawdzając pierwszy znak, aby określić, czy jest to polecenie trybu bezpośredniego. Ponieważ pierwszy znak jest liczbą, linia zostanie potraktowana jako nowa, a Basic usunie zmienne wskaźniki do ich wartości domyślnych i przechowa nową linię w pamięci.

Przypuśćmy, że później zostanie wprowadzone polecenie GOTO 90. Gdy znak G zostanie wprowadzony z klawiatury, procedura sprawdzi koniec obszaru wskaźnika zmiennej i ustali, że jego domyślna wartość jest mnieisza niż wartość zapisana. Powoduje to op-RESTORE. która ustawia erację wskaźniki z powrotem na ich oryginalne (czyli poprawne) wartości. W związku z tym, zmienne zostaną wyświetlone tak, jakby nie wprowadzono żadnych zmian w programie.

Aby jeszcze bardziej skomplikować sprawę, należy wziąć pod uwagę rezultat działania przycisku RESET. Jak wspomniałem wcześniej, procedura jest wywoływana za pośrednictwem wektora KSW (adres \$38-\$39). Applesoft wykonuje pośredni skok do adresu znajdującego się w tym wektorze, gdy wymagane jest użycie klawiatury. Nasza procedura ustawia KSW tak, że będzie wskazywał na siebie zamiast zwykłej procedury wejściowej KEYIN.

Po naciśnięciu przycisku RESET następuje szereg operacji, które między innymi inicjalizują wyjście wideo oraz wektory wejścia-wyjścia. Sekwencja RESET kończy się skokiem do adresu utrzymywanego w wektorze SOFTEV (\$3F2-\$3F3), który zwraca kontrolę do Basica. W ramach tej sekwencji wektor KSW zostaje zmieniony na jego wartość domyślną (wskazującą na KEYIN), tym samym wyłączając nowy podprogram.

Aby temu przeciwdziałać, wektor SOFTEV jest ustawiamy tak, aby spowodować skok do własnej procedury resetowania, która ponownie uaktywnia procedurę, a następnie wychodzi normalnie do Basica.

Jak już wspomniałem, tylko wskaźniki, a nie same zmienne, są tracone podczas wprowadzania zmian w programie. Jest to prawdą tylko wtedy, gdy zmiana nie powoduje wydłużenia objętości programu. Jeśli program zostanie wydłużony, dolny koniec tabeli zmiennych zostanie nadpisany przez linie programu i trwale utracony.

Tego problemu można łatwo uniknąć, używając LOMEM do ustalenia przestrzeni w pamięci między końcem programu a początkiem tabel zmiennych. Ta przestrzeń jest następnie dostępna dla dodatkowej linii programu bez zakłócania zmiennych. Przestrzeń 256 bajtów jest odpowiednia dla około ośmiu linii w Basicu i taki obszar może być łatwo przydzielony przez użycie następującej instrukcji:

1 LOMEM : PEEK (105) + PEEK (106) * 256 + 256

Musi być ona wpisana jako pierwsza linii programu. Jeśli przewidujesz wprowadzanie wielu zmian, obszar można powiększyć, zwiększając ostatnią wartość w tej linii. W bardziej zwarty sposób można to wpisać tak:

1 POKE 106, PEEK (106) + 1

I znowu, przestrzeń można zwiększyć o 256 bajtów, zwiększając ostatnią wartość w linii.

Po załadowaniu procedury do pamięci, uruchom program, aby ustawić wektory KSW i SOFTEV. Przedstawiony listing można uruchomić za pomocą CALL 755. Załaduj lub wprowadź program w Basicu i przed uruchomieniem wpisz instrukcję CLEAR, a naciśnij kombinację CTRL-C, aby zainicjować rejestry pamięci.

STREFX XPPLE

Następnie ustaw LOMEM co najmniej kilkaset bajtów poza końcem programu, aby zwolnić miejsce w pamięci dla nowych linii programu. Można to zrobić, wprowadzając LOMEM z klawiatury lub wprowadzając do programu jedną z instrukcji zawartych w poprzednim akapicie.

Uruchom program jak zwykle, poprzez RUN lub GOTO. Można go dowolnie przerwać lub zmodyfikować, a działanie będzie kontynuowane bez utraty zmiennych. Problemy mogą wystąpić, jeśli zostaną wprowadzone polecenia, które zmieniają tabele zmiennych lub ich wskaźniki.

Zmiana HIMEM lub LOMEM może skutkować tym samym. Zmiana LOMEM nie sprawi problemów, chyba że nastąpi po CTRL-C. W takim wypadku utracone zostaną wszystkie zmienne. Dodajmy, że zmiana HIMEM wpłynie tylko na ciągi tekstowe.

Oczywiście, zmiana HIMEM lub LOMEM może zniszczyć wartości zmiennych, niezależnie od tego, czy nowa procedura jest w użyciu, czy też nie, więc tych poleceń nigdy nie należy używać po przypisaniu zmiennych do programu. Inny problem może wystąpić, jeśli program jest uruchamiany, gdy obszar przechowywania wskaźnika podprogramu ma nieprawidłową zawartość lub wartości wskaźnika przypisane do innego programu.

Dlatego przed uruchomieniem programu powinna być zawsze stosowana opisana powyżej sekwencja zawierająca instrukcję CLEAR i kombinację klawiszy CTRL-C.

Można to również zrobić automatycznie, umieszczając następującą linię:

2 CLEAR: CALL 808

na początku każdego programu. Mając na uwadze te proste środki ostrożności,

Program numer 1

02F0"	00	00	00	A9	45	AØ	03	80
02FB"	F2	03	BC	FЗ	03	20	6F	FB
0300"	A9	0 B	AØ	03	85	38	84	39
0308"	4C	EA	03	84	F9	20	IB	FD
0310"	85	FA	C 9	83	FØ	12	A5	6E
0318"	CD	50	03	90	18	DØ	09	A5
0320"	6D	CD	4F	03	90	ØF	FØ	18
0328"	A0	07	B9	69	00	99	4 B	03
0330"	88	10	F7	30	0 B	A0	07	B9
0338"	4 B	03	99	69	00	88	10	F7
0340"	A4	F 9	A5	FA	60	20	00	03
0348"	4C	BF	9D	00	00	00	00	00

```
Program numer 2
 10 REM TEST PROCEDURY
             PEEK (105) + PEEK (106) * 256
 20 LOMEM:
    + 256
 30 CLEAR : CALL 808: REM ZAPISUJEMY
    PROCEDURE"(ADRES $328)
 40 A " 1:B = 2:C = 3
 50 A$ = "A":B$ = "B":C*$ = "C"
 60 \text{ FOR I} = 1 \text{ TO } 10
 70 \text{ ARRAY(I)} = I
 80 NEXT I
 90 FOR I = 1 TO 10
100 PRINT ARRAY(I)
110 NESTI: REM ZWROC UWAGE NA TO
120 PRINT
130 PRINT A, B, C
140 PRINT
150 PRINT A$, B$, C$
160 REM SPROBUJ ZMIENIC TE LINIE
```

przedstawiona procedura może sprawić, że programowanie i obsługa błędów w Basicu będzie znacznie szybsza i wygodniejsza, szczególnie przy pisaniu dłuższych programów.

Karty turbo do Apple II

Komputery 8-bitowe raczej nie możliwości znane Ζ sa przyspieszenia procesora. Co powstawały takie prawda karty, ale mało kto je kupował, wielu wypadkach bo w powodowały problemy z kompatybilnością. Apple II pojawił się na rynku w 1977 roku, a już ok. 5 lat później zaczęto mówić o możliwościach jego przyspieszenia.

W 1982 roku firma Number Nine Computer Corporation opracowała akcelerator dla Apple II i Apple II Plus. Karta o nazwie Apple Booster kosztowała 600 dolarów i zwiększała prędkość od 1 MHz do 3,58 MHz. Ciekawostką jest fakt, iż karta używała 64 kilobajtów pamięci RAM, aby stworzyć nowe środowisko pracy, natomiast reszta komponentów była potrzeba tylko jako urządzenia wejścia-wyjścia. Przypomina to dzisiejsze karty opracowywane w technologii FPGA, które traktują macierzysty komputer jako terminal.

Inną kartą tego samego typu był Speed Demon, który pojawił się w sprzedaży w 1983 roku za cenę ok. 300 dolarów. Wykorzystał procesor 65c02 działający z częstotliwością 3,58 MHz i wykorzystywał 4 kilobajty pamięci jako pamięć podręczna (ang. cache). Pozwoliło to uzyskać szybszy czas dostępu, niż do wolniejszej pamięci w Apple II, dzięki czemu wiele programów pracowało szybciej.

W 1983 roku akcelerator firmy Number Nine Corporation ponownie pojawił się na rynku, ale tym razem był sprzedawany w ulepszonej formie pod nazwą Accelerator II. Podobnie jak pierwowzór, karta miała 64 kilobajty pamięci RAM, ale tym razem można było ją skonfigurować tak, aby w razie potrzeby przywrócić normalną prędkość działania. Chodziło oczywiście o zachowanie zgodności z oprogramowaniem, które nie chciało działać lub pracowało nieprawidłowo z aktywnym przyspieszeniem.

Accelerator II działał na Apple II i II Plus, ale powodował problemy na modelu Apple IIe. Ponadto firma musiała zmienić swoją nazwę ze względu na spór prawny, o którym niewieie wiadomo. W każdym razie pod nową nazwą Titan Technologies w 1984 roku wydano kartę Accelerator IIe. Wykorzystała ona procesor 65c02 i dodała kolejne 16 kilobajty pamięci RAM, do której skopiowany był kod pamięci ROM komputera. Znowu celem było szybsze działanie.

W 1986 roku firma Applied Engineering wprowadziła na rynek płytę akcelerator TransWarp. Produkt ten był sprzedawany dłużej na rynku niż jakikolwiek inny, prawdopodobnie dlatego, że zdecydowano się na dużo lepszą reklamę. TransWarp był dostępny z procesorem 65c02 i 256 kilobajtami pamięci o zwiększonej prędkości. Obszar 256 KB pamięci RAM został podzielony na cztery banki po 64 KB. Pierwszy bank posiadał zapisaną zawartości kości ROM komputera, drugi utrzymywał kopię pamięci RAM z płyty głównej, a trzeci i czwarty bank był przeznaczony na pracę jako rozszerzenie pamięci.

Karta mogła być również kontrolowana w pewnym zakresie za pomocą dołączonego oprogramowania, była kompatybilna wcześniej 7 wydanymi akceleratorami, ponato można było zamieniać prędkość za pomocą manipulacji wartością zapisaną pod adresem \$C074. Było to bardzo ciekawe rozwiązaniem, po raz kolejny przypominające dużo późniejsze "podkręcanie" stosowane w epoce popularności pecetów.

mojego krótkiego omówienia Ζ wyraźnie widać, jak poważnie był traktowany komputer Apple II na rynku amerykańskim. Nasze domowe Commodore czy Atari również miały możliwość rozbudowy, ale karty turbo rozpowszechniły się później, gdy korzystaliśmy z modeli 16-bitowych. Rynek w USA był zdecydowanie inny niż w Europie, nie mówiąc o naszym kraju, dlatego podczas dyskusji w sieci historii komputerów proponuję 0 spojrzeć szerzej, niż tylko biorąc pod uwagę sprzęt, który akurat mieliśmy jako jedyny słuszny punkt wyjścia..

> Opracował: Marcin Libicki



Acorn A3010

Acorn nazwę dla swojej marki ponoć z dwóch wybrał powodów, po pierwsze dlatego, że chciał pojawić się powyżej Apple w książce telefonicznej, ро drugie, а ponieważ nazwa oznaczała rozwijającą się firmę - Acorn w języku angielskim to żołądź.

I rzeczywiście tak było. Firma została założona w 1978 roku i choć jej historia była dość burzliwa, odnosiła niemałe sukcesy na rynku komputerów domowych oraz naszym edukacyjnych. Przykładem jest BBC Micro, czyli popularny sprzęt stworzony w celu używania w szkołach. W 1983 roku Acorn rozpoczął pracę nad procesorem ARM, który znamy dzisiaj z wszechobecnych mobilnych. urzadzeń Można tu powiedzieć jako ciekawostkę, że ARM początkowo miał byc ulepszoną wersją procesora 6502 znanego między innymi z 8-bitowych komputerów Commodore. Celem projektu było opracowanie nowego potężniejszego mikroprocesora.

Udało się to bardzo dobrze, bowiem ARM utrzymywał firmę przy życiu bardzo długo przed upowszechnieniem się smartfonów i tabletów. W 1987 roku Acorn wyprodukować komputer o oznaczeniu A300 wyposażony w 8bitowy dźwięk stereo i 256 kolorów. Sprzedaż wyglądała bardzo dobrze w Irlandii, Australii i Wielkiej Brytanii. Dwa lata później stworzono serię 3000, która wykorzystała uniwersalny typ obudowy z wbudowaną klawiaturą zgodną z ówczesnymi trendami, którymi najpopularniejszymi przedstawicielami były komputery Atari ST i Amiga.

Archimedes A3000 wyposażony był w procesor ARM2 z zegarem 8 MHz, 1 MB pamięci RAM i dostarczano go z nowym systemem operacyjnym RISC 2. Mógł pracować a nowymi rozdzielczościami ekranu, nawet 800x600 przy 16 kolorach.

W 1992 roku wprowadzono na rynek model A3010, który miał być komputbędzie erem domowym, który konkurował z ofertami firm Commodore i Atari. Został wyposażony w modulator telewizyjny umożliwiający korzystanie ze standardowego gniazda RF, 9pinowe złącza joysticka, a także nowy procesor ARM250 z szybszym zegarem 12 MHz i zintegrowanymi układami dźwiękowymi i graficznymi. Biorąc pod uwage, że urządzenie to zawierało również nowy i szybki system RISC

OS, była to dość potężna maszyna, która pod wieloma względami mogła bez problemu konkurować z rozbudowaną Amiga.

Zastosowany procesor ARM to RISC, co oznacza Reduced Instruction Set Computing. Ogólnie rzecz biorąc, mamy mniej funkcji wbudowanych w procesor, dzięki czemu jest on bardziej wydajny i pobiera mniej prądu. Pierwotnie Acorn planował użyć technologii RISC do maszyn biurowych, ale szybko okazało się, że może być z powodzeniem wykorszystana jako podstawa komputerów Archimedes.

W porównaniu do CISC, które mają złożony zestaw instrukcji, Archimedes A3010 był kilkakrotnie szybszy od procesora CISC (złozony zestaw instrukcji) Motorola 68000 obecnego w Amidze i Atari ST. Widać to bardzo dobrze na przykład podcas generowania fraktali



Mandelbrota czy grach 3D. Wyjątkami są sytuacje, kiedy nalezy wykonać poszczególne skomplikowane instrukcje, co wymagało wielu kroków dla procesorów ARM, natomiast działa w jednym kroku na modelu 68000.

Inną mocną stroną Archimedesa jest system operacyjny wbudowany w ROM. W komputerze A3010 mamy RISC OS 3.11, który wykorzystuje wielozadaniowość i jest niezwykle przyjazny dla nawet początkującego użytkownika. System był dostarczany w pakiecie z wieloma programami, które ułatwiają korzystanie z komputera i obsługują także BBC Basic, co daje możliwość nauki programowania jak na klasycznym komputerze z lat '80-tych.

Poza tym urządzenie wygląda bardzo podobnie do innych komputerów domowych z tamtego okresu, chociaż Acorn próbował zachować czerwony klawisz funkcyjny na klawiaturze. Aby wyróżnić serię w zależności od ich zastosowań, przyjęto zielony schemat kolorów dla domowych serii A3010, czerwony dla modeli edukacyjnych, takich jak oryginalny A3000 i A3020 (został wydany w tym samym czasie co model A3010, ale z większą ilością pamięci i twardym dyskiem dysk) i szary dla maszyn przeznaczonych do użytku biznesowego, takich jak A4000 (również sprzedawany w tym samym okresie).

Klawiatura w A3010 jest bardzo wysokiej jakości, dla wielu osób może wydawać się lepsza niż w Amidze i Atari ST. Z tyłu obudowy wamy następujące porty:

- równoległy,
- szeregowy,
- port joysticka 1,
- port joysticka 2,
- port myszy,
- gniazdo słuchawkowe stereo,
- gniazdo monitora,
- wyjście wideo dla monitora,
- wyjście wideo dla telewizora (RF).



Komputer Acorn A3010 po zdjęciu górnej części obudowy.

Zasilacz wbudowany jest w obudowę, a z tyłu dostępny jest przełącznik zasilania, podobnie jak w Atari ST. Po lewej stronie znajduje się przycisk Reset. 3,5calowa stacja dyskieek pozwala sformatować dyskietki HD i używać pojemności 1,6 MB. Standardowo komputer jest wyposażony w 1 MB pamięci, którą można łatwo rozszerzyć do 2 MB.

Archimedes A3010 może generować 8kanałowy 8-bitowy dźwięk stereo, a także wyświetlać kilka rozdzielczości do 800x600. Mamy maksymalnie 256 kolorów jednocześnie na ekranie, z palety 4096 barw. Jak się dzisiaj okazuje, układ graficzny może obsłużyć rozdzielczości nawet do 1152×896 (w 2 kolorach), ale możliwości te nie były obsługiwane w systemie operacyjnym. Obraz Acorna nie zawsze posiada typowe częstotliwości odświeżania, dlatego nie wszystkie monitory VGA będą działać prawidłowo. Najlepiej oczywiście posiadać prawdziwy monitor Multisync.

Biblioteka gier dla serii komputerów Acorn nie jest tak bogata jak dla Amigi czy Atari, ale istnieje wiele słynnych tytułów jak Zool, Syndicate, Descent, Diggers, Lotus, Last Ninja, Lemmings, Alone in the Dark czy Flashback. Można więc zagrać w ciekawe produkcje, a dzięki szybkiemu procesorowi symulatory operujące na grafice wektorowej działają płynnie i rozgrywka jest dynamiczna. Taką szybkość pracy można uzyskać dopiero na Amidze 1200, najlepiej z dodatkową pamięcią.

Po A3010 pojawił się model A7000 z 1995 roku, który jest ostatnim komputerem z serii Archimedes. Ten sprzęt jest bardziej podobny do komputerów RISC PC, które zostały wprowadzone na rynek jako następca starszej serii. A7000 posiada procesor z zegarem 32 MHz o obsługuje nawet 128 MB pamięci RAM.

Podsumowując, seria Archimedes była bardzo udana, a model A3010 byłby większym hitem, gdyby został wprowadzony rynek nieco na wcześniej. Jednak architektura ARM przetrwała do dziś i jest bardzo popularna, co jest prawdziwym osiągnięciem firmy, która później przekształciła się w ARM Holding. W roku 2005 wyprodukowano blisko 1,7 miliarda procesorów opartych na technologii ARM, co stanowiło ok. 75% wszystkich 32bitowych procesorów dostępnych na rynku. Co ciekawe, w 2016 roku spółka została przejęta przez japońską firmę SoftBank, ale to zupełnie inna historia.

Programowanie w systemie RiscOs część 1.

Jeżeli chcecie pisać programy dla systemu RISC OS, które będą korzystały z interfejsu graficznego, musicie koniecznie przeczytać mój artykuł. Nie jest to czysty samouczek, ale opowiem jak używać okien, menu, systemowego pulpitu i wielozadaniowości. Pokażę jak opanować system o nazwie WIMP, czyli Windows (okna), Icons (ikony), Menu, Pointers (wskaźniki).

Na początek mała uwaga - przyjmuję, że Czytelnik posiada podstawową znajomość Basica, poleceń, składni, pętli typu FOR i instrukcji warunkowych. Nie będę zaczynał od "elementarnych podstaw" - jak mówiła moja nauczycielka matematyki w liceum, ale przedstawię najkrócej możliwości korzystania z ciekawych funkcji.

Pierwszym ważnym krokiem jest stworzenie powłoki WIMP, która będzie zawierać wszystkie procedury potrzebne do tworzenia okien i menu oraz radzenia sobie z działaniami użytkownika, takimi jak kliknięcia myszy. Możemy wtedy dodać działającą własną aplikację. Wszystko można wykonać dość szybko, a na dodatek nie trzeba znać kodu maszynowego, ani innych języków programowania za wyjątkiem Basica. Nie ma on jednak żadnych poleceń do kontrolowania okien i menu, dlatego będziemy wykonywać częste wywołania różnych poleceń systemu operacyjnego.

Są one nazywane "Software Interrupts", w skrócie "SWI" i należy ich używać za pomocą podstawowego polecenia SYS. Istnieje bardzo wiele SWI, w tym ponad 50 odnosi się tylko do WIMP-a, choć na razie nie będziemy używać więcej niż kilku z nich. Stopień rozbudowania tego systemu pokazuje objętość oryginalnego podręcznika programisty systemu RISC OS, który zawiera 5 tomów i ponad 4000 stron.

Każdy SWI ma numer, a także nazwę, która składa się z dwóch części. Pierwsza odnosi się do sekcji systemu operacyjnego, która obsługuje wywołanie - "OS" dla jądra lub głównej części, "WIMP" dla menedżera okien itp. Druga część to nazwa samego wywołania. Te dwie części są oddzielone symbolem podkreślenia ("_").

Bardzo ważne jest, aby nazwa była wpisania bardzo dokładnie, w przeciwnym razie system operacyjny jej nie rozpozna. Wszystkie słowa w drugiej części nazwy są ze sobą połączone, a każde zaczyna się od wielkiej litery, podobnie jak słowo w pierwszej części. Cała reszta nazwy pisana jest małymi literami, z wyjątkiem przypadków, gdy część nazwy zawiera skróty, tak jak w "OS" lub "CLI", które oznaczają "System operacyjny" i "Interpreter inii poleceń". Przykładem jest nazwa:

Wimp_GetPointerInfo

Procesor naszego komputera ma 16 rejestrów, z których każdy może pomieścić 32 bity lub cztery bajty. Są one znane jako symbole od R0 do R15. W artykule będzie korzystać tylko z pierwszych sześciu rejestrów, ale przekazywanie informacji pomiędzy pozostałymi należy wykonywać w analogiczny sposób.

Uruchom Basic w trybie bezpośrednim, czyli naciśnij klawisz F12. Dalej wpisz poniższą linię:

SYS "OS_WriteC",65

Polecenie SYS wywołuje SWI o nazwie "OS_WriteC", który jest odpowiednikiem podstawowego polecenia "VDU". Zanim to zrobi, umieszcza liczbę (wpisaną po przecinku, czyli 65), w rejestrze R0. "OS_WriteC" to wywołanie systemowe, które pobiera liczbę z rejestru R0 i wyświetla rezultat działania na ekranie. Dlatego po naciśnięciu klawisza ENTER powinieneś zobaczyć literę "A". Tak się dzieje, bowiem wartość 65 to kod ASCII właściwy właśnie dla znaku "A". Jeśli zamiast tego na ekranie pojawi się poniższy komunikat o błędzie:

SWI name not known

oznacza to, że SWI nie zostało rozpoznane ze względu na błędnie wpisaną nazwę.

Teraz spróbuj pisać bardziej skomplikowaną linię:

SYS "OS_ReadC" TO a%

Po naciśnięciu ENTER znak zachęty w Basicu nie pojawi się ponownie, dopóki nie naciśniesz innego klawisza. Jeśli teraz wpiszesz:

PRINT CHR\$a%

zobaczysz znak związany z klawiszem, który został naciśnięty. Jak można się domyślić, "OS_ReadC" jest odpowiednikiem podstawowego polecenia GET. Czeka na naciśnięcie klawisza, a następnie umieszcza kod ASCII dla klawisza w rejestrze R0. Komenda SYS pobiera numer z tego samego rejestru i umieszcza go w zmiennej "a%" (cały czas w Basicu).

Możemy też wpisać polecenie takie jak poniżej:

SYS "Xxx_XxxXXxxx",x% y%,z% TO a%,b%,c%

Linia tapobiera zmienne x%, y% i z% i umieszcza je odpowiednio w rejestrach - R0, R1 i R2. Następnie wywołuje SWI, a po powrocie liczby z R0, R1 i R2 zapisuje w zmiennych, które występują po słowie kluczowym TO, czyli w tym przypadku - a%, b% i c%.

Typowe polecenie używane przez WIMP to:

SYS "Wimp_CreateIcon",,b% TO i%

Zwróć uwagę, że po nazwie SWI są dwa przecinki, co powoduje, że instrukcja przyjmuje wartość b% i umieszcza ją w rejestrze R1. Liczba zero jest też umieszczana w rejestrze R0, ale w tym momencie nie ma to znaczenia. Po powrocie z SWI liczba z R0 jest zapisywana jako i%.

Ważnym aspektem aplikacji opartych na WIMP jest fakt, że są wielozadaniowe, co oznacza, że mogą działać w tym samym czasie, co inne programy. W najprostszym przypadku oznacza to, że można załadować program i używać go bez konieczności zamykania innego, wcześniej uruchomionego. Ułatwia to również przesyłanie danych między programami i korzystanie z kilku programów jednocześnie.

Bardzo łatwo można napisać prosty i krótki program wielozadaniowy, a nawet umieścić go w tylko jednej linii w Basicu. Oto przykład: na osobne linie. Będzie to wyglądało tak jak w ramce na dole strony. Dla ułatwienia niektóre części rozdzieliłem odstępami.

Numery linii nie mają znaczenia i zostały wpisane wyłącznie w celu ułatwienia prezentacji działania programu. W zależności od edytora, którego używasz do wpisywania programów, możesz zauważyć, że linie są numerowane kolejno co jeden numer, a nie co 10, ale to tylko techniczna zmiana, bez znaczenia odnośnie funkcjonalności programu.

Ze względu na ograniczoną objętość miejsca w naszym magazynie, działanie programu omówię w następnym odcinku. Zachęcam też do samodzielnej analizy i "rozgryzania" listingu, bo nie jest on bardzo skomplikowany, a pozwala zrozumieć działanie wielu funkcji.

Opracował: Mariusz Wasilewski

DIM b% 255:SYS "Wimp_Initialise",200,&4B534154,"MyApp": t%=TIME:REPEAT SYS "Wimp_Poll",,b%:WHILE TIME>t%+100: VDU 7:t%+=100:ENDWHILE:UNTIL FALSE

Zauważ, że całość powinna być wpisana jako jedna linia. Jednak nie bedziesz mógł uruchomić tego programu bezpośrednio w wierszu poleceń. Musisz wpisać go w edytorze tekstu, zapisać jako plik Basica i dopiero w takiej formie uruchomić, klikając dwukrotnie na ikonę pliku.

Zanim uruchomisz program, warto wiedzieć, jak będzie on działać. Najłatwiej to zrobić, dzieląc listing 10 DIM b% 255

- 20 SYS "Wimp_Intialise",200,&4B534154, "MyApp"
- 30 t%=TIME
- **40 REPEAT**
- 50 SYS "Wimp_Poll",,b%
- 60 WHILE TIME>t%+100
- 70 VDU 7
- 80 t%+=100
- 90 ENDWHILE

100 UNTIL FALSE 110 Rem XXX KONIEC GLOWNEJ PETLI XXX

Kolory w Basicu

Jedną z głównych nowych cech Archimedesa jest możliwość stosowania palety 4096 różnych kolorów. Zapewnia to ogromny postęp, szczególnie w porównaniu z sześcioma kolorami plus czarny i biały, które dostępne były wcześniej. Oczywiście wszystko można poprawić i to samo dotyczy użycia koloru na Archimedesie. Na ekranie możemy mieć jednocześnie tylko 256 kolorów, ponadto decyzja Acorna o zachowaniu pełnej kompatybilności Ζ wcześniejszym sprzętem doprowadziła do dodatkowego zamieszania.

Istnieją dwa różne podejścia do obsługi kolorów na Archimedesie. Przede wszystkim należy spojrzeć na tryby od 0 do 7 używane jeszcze w BBC Micro oraz wszystkie inne tryby, które pozwalają na użycie maksymalnie 16 kolorów na ekranie w tym samym czasie. Kolory mogą być dowolnie wybrane spośród wszystkich dostępnych 4096 odcieni, które potrafi wygenerować Archimedes.

Sposób ich użycia jest bardzo podobny używanych w poprzednich wersjach podstawowego modelu komputera BBC Micro, choć w Polsce pewnie prawie nikt go nie zna. Powiedzmy więc, że tryby o numerach 10, 13 i 15 umożliwiają wyświetlanie do 256 kolorów na ekranie w tym samym czasie, ale metoda korzystania z nich jest zupełnie inna niż w pozostałych trybach.

Maksymalna ilość kolorów w zależności od trybu graficznego

Ilość kolorówNumery trybów20,3,4,6,18,23,2541,5,8,11,19,26162,7,9,12,14,16,17,20,2725610,13,15,21,24,28

Listę trybów oraz dostępnych kolorów przedstawiam w ramce. Te same zasady obowiązują dla trybów 2, 4 i 16. W każdym z tych trybów wybór koloru odnosi się do wartości logicznej. Na przykład w trybie 1 wartości te wynoszą od 0 do 3, a w trybie 12 - od 0 do 15. Z każdym numerem logicznym związany jest kolor fizyczny, który faktycznie widzimy na ekranie. Szczegóły widać w drugiej ramce.

W dowolnym momencie aktywne będą cztery kolory. Są to: bieżące kolory tekstu pierwszego

tła planu, oraz kolory grafiki na pierwszym planie i w tle. W przypadku tekstu i grafiki kolor tła jest wskazywany przez dodanie liczby 128 do logicznego numeru koloru. Odbywa się to za pomoca instrukcji COLOUR dla tekstu i GCOL dla grafiki.

Przy domyślnych kolorach w trybie 12, będzie to wyglądało tak:

COLOUR 2

Powyższa linia ustawi kolor zielony dla tekstu. Jeśli wprowadzimy taką linię:

COLOUR 128+6

wskażemy kolor jasno niebieski (Cyjan) jako nowy kolor tła. Dodajmy, że użycie instrukcji COLOR w ten sposób określa kolor pierwszego planu i tła dla dowolnego tekstu wyświetlanego później na ekranie. Nie zmienia to kolorów już wyświetlonego tekstu.

Domyślne przypisania kolorów dla trybów 16-kolorowych.

Liczba	Kolor
0	Czarny
1	Czerwony
2	Zielony
3	Żółty
4	Niebieski
5	Magenta
6	Cyjan
7	Biały
8	Migający czarny lub biały
9	Migający czerwony lub cyjan
10	Migający zielony lub purpurowy
11	Migający żółto-niebieski
12	Migający niebieski lub żółty
13	Migający magenta lub zielony
14	Migający cyjan lub czerwony
15	Migający biały lub czarny



Grafika w formacie PNG

Wśród najpopularniejszych formatów graficznych jest PNG, który dzisiaj jest szeroko stosowany w sieci. System Amigi może go obsługiwać, o ile zostanie zainstalowany tak zwany "datatyp", czyli pliki dodające do Workbencha możliwość rozpoznawania określonego formatu.

Polecam korzystać z niego za pomocą pakietu o nazwie "Warp PNG", którego najnowsza wersja datowana jest na luty 2015 roku. Obsługuje on wszystkie procesory serii Motorola 68000, a także PowerPC, które omawiamy w osobnym rozdziale. Na "Warp PNG" warto zwrócić uwagę także dlatego, że obsługuje tryb "Interlace", dzięki któremu grafika może być pokazywana nawet jeśli nie kompletna, bez użycia iest dodatkowych specjalistycznych funkcji.

Nazwa trybu nie ma wiele wspólnego z wyświetlaniem obrazu z przeplotem i nie należy go mylić. Jest on jednak bardzo przydatny na przykład przy pobieraniu dużych plików. Ponadto możliwe jest korzystanie z tak zwanego kanału "alfa" stosowanego dla uzyskania efektu przezroczystości oraz korekcji "gamma" pozwalającej dostosować wyświetlanie grafiki do określonego monitora. Funkcje te są ważniejsze niż może Ci się to

wydawać, bowiem gdyby datatyp ich nie obsługiwał, wiele plików aktualnie dostępnych w Internecie nie mogłoby być w ogóle wyświetlonych.

Aby dodać obsługę formatu PNG do Workbencha, a tym samym do programu "CyberView", przejdź na Aminet - tym razem do znajomego już katalogu "util/dtype". Pobierz plik o nazwie "WarpPNGdt.lha" i rozpakuj go w standardowy sposób do "Ram Dysku":

lha x WarpPNGdt.lha RAM:

lub do innego katalogu na dysku twardym, zamieniając "RAM:", tak jak mówiliśmy o tym wiele razy wcześniej. W katalogu docelowym znajdzie się ikona podpisana jako "WarpPNGdt". Najedź na nią wskaźnikiem i naciśnij dwa razy lewy klawisz myszki. Pojawi się nowe okno, w którym również wykonaj "dwuklik", lecz teraz na pozycji o nazwie "Install_WarpPNG".

Uruchomisz w ten sposób typowy program instalacyjny. Pierwsza informacja to wybór trybu instalacji. Domyślnie wskazana jest opcja dla użytkownika "średniozaawansowane- go". Nie zmieniaj tego, lecz skorzystaj z przycisku "Kontynuuj instalację" (ang. "Proceed With Install") na dole. Następne pojawią się opcje instalacji, a w trzecim etapie – informacje na temat koniecznej rejestracji pakietu. W obu przypadkach nie należy modyfikować żadnych funkcji, a tylko znowu wskazać pole "Kontynuuj" (ang. "Proceed").

Dodajmy, że "Warp PNG" należy do oprogramowania typu Shareware i autor oczekuje uiszczenia opłaty rejestracyjnej, jednak - jak sam dodał w dokumentacji – nie ograniczył funkcji swojego produktu, aby każdy mógł samodzielnie przekonać się do wsparcia. Tak więc opłata zależy od naszych dobrych chęci i docenienia autora, do czego oczywiście zachę- cam.

Wszystkie komunikaty są w języku polskim, co ułatwia obsługę. Musisz tu wybrać procesor jaki posiadasz w swojej Amidze. Program stara się automatycznie określić właściwy symbol, ale powinieneś to zwery- fikować. W naszym przypadku wszystko zostało rozpoznane prawi- dłowo, faktycznie posiadamy procesor Motorola 68040. Gdy upewnisz się, że aktywna jest poprawna opcja, ponownie wybierz przycisk "Kontynuuj".

Następne dwa etapy instalacji polegają na potwierdzeniu skopiowania składników umożliwiających funkcjonowanie mechanizmu rozpoznawania nowego formatu danych. Oba będą wyglądały podobnie, jak poniżej:

Jedyną różnicą będzie ścieżka dostępu widoczna w polu "Wybrany Katalog". Opcje sugerowane przez program są prawidłowe, dlatego bez wykonywania



zmian wybierz dwukrotnie przycisk "Kontynuuj". Zwróć uwagę, że w polu z nazwa katalogu jest od razu aktywny kursor, dlatego trzeba uważać, aby nie wpisać przypadkowo żadnych znaków. Inaczej instalacja się nie powiedzie. Wskazane ścieżki dostępu to typowe miejsca w systemie, gdzie zapisywane są pliki związane z datatypami. W zasadzie nie musisz się nimi przejmować, ale warto je znać. Więcej na ten temat dowiesz się z pierwszego tomu podręcznika, z rozdziałów zatytułowanych "Dysk systemowy" oraz "Formaty danych". Instalator będzie teraz kopiować niezbędne pliki, ale powinno to potrwać tvlko kilka sekund. W trakcie operacji widoczny będzie napis:

Instalacja w toku

Może być on jednak trudny do zauważenia, bo szybko zastąpi go dłuższa informacja o zakończeniu całego procesu. Ponownie dwukrotnie wybierz przycisk "Kontynuuj".

W pierwszym oknie zostaniesz poinformowany o tym, że należy wykonać reset komputera: Widać tu też ostrzeżenie, że jeśli masz zainstalowane inne datatypy obsługujące format PNG - musisz je skasować. Mimo to zwykle nie należy nic zmieniać, chyba że po ponownym wczytaniu Workbencha nowe pliki nie będą funkcjonować poprawnie. Ostatnia informacja powinna brzmieć:

Instalacja zostala zakonczona

Gdy okno programu zniknie, po prostu zresetuj Amigę. Jeżeli stosowałeś się do naszych wcześniejszych wskazówek, wszystko będzie działać bez przeszkód. System powinien załadować się tak jak poprzednio, bez dodatkowych komunikatów. Aby sprawdzić, czy format PNG będzie obsługiwany możesz uruchomić przeglądarkę "Multiview", a także "CyberView". W zasadzie nie ma znaczenia, który program wywołasz, bo oba korzystaja z datatypów. Dla pewności możesz spróbować wyświetlić grafikę w jednym i drugim. Wykonaj to analogicznie do wcześniejszych wskazówek, czyli wskaż plik z rozszerzeniem ".png" w oknie wyboru. Wiele przykładowych obrazów znajdziesz w Internecie, tylko, pamietaj że powinny być odpowiednie dla ilości pamięci w Twoim komputerze. Nie staraj się pobrać jak największych plików, bo możesz mieć problemy z ich odczytaniem.

Proponujemy przeprowadzić test na pliku w rozmiarze od kilkudziesięciu do kilkuset kilobajtów, a dopiero później próbować wczytywać większe obrazy. Wyświetlanie grafiki typu PNG przebiega podobnie jak w przypadku innych formatów, jednak nasz nowy datatyp charakteryzuje się jedną specyficzną cechą. Otóż dane są najpierw poddawane analizie i dekompresji, a dopiero później pokazywane na ekranie monitora. Dlatego, gdy wskażesz plik w oknie wyboru, musisz poczekać kilka lub kilkanaście sekund, w trakcie których pozornie nic się nie dzieje.

Możesz jednak zauważyć zwolnienie pracy Workbencha, co jest spowodowane odczytywaniem informacji z pliku. Pamiętaj, że czas oczekiwania zależy od szybkości procesora i wielkości pliku, dlatego może to trwać krócej lub dużo dłużej niż podajemy. Poczekaj, a gdy program zakończy analizę całego pliku - grafika zostanie wyświetlona. Pozostałe funkcje obu przeglądarek – zarówno "Multiview", jak i "CyberView" - będą działać bez zmian. Zwróć też uwagę, że pierwsza z nich domyślnie pokazuje obraz w oknie, a nie na osobnym ekranie, więc jakość będzie uzależniona od trybu wyświetlania ustawionego na Workbenchu.

Datatyp "Warp PNG" posiada ustawienia, które możesz kontrolować za pomocą programu zapisywanego w katalogu "Prefs" na dysku systemowym. Niestety wymaga on systemu operacyjnego w wersji 3.5 lub nowszej, dlatego w naszym przypadku jest bezużyteczny. Autor przewidział jednak ręczną zmianę "preferencji" za pomocą dodatkowej zmiennej środowiskowej. Teraz zajmiemy się obsługą zmiennej "WarpPNG.prefs", która pozwala wpływać na mechanizmy rozpoznawania i wyświetlania plików w formacie PNG na Workbenchu.

W pierwszej kolejności musisz utworzyć zmienną, w ramach której będziesz zapisywać ustawienia. W tym celu otwórz okno "Shell" i wpisz w nim:

setenv ENV:Datatypes/Warp-PNG.prefs IMAGE_GAMMA 1500000

lub krócej:

setenv Datatypes/Warp-PNG.prefs IMAGE_GAMMA 1500000

Obie wymienione linie są równorzędne. Rozpoczyna je polecenie SETENV, następnie podajemy nazwę pliku, pod którą umieszczona będzie zmienna. Ostatni fragment to treść jaka będzie zapisana i to właśnie ona określa parametry wyświetlania plików w formacie PNG.

Nazwę urządzenia logicznego "ENV:" możesz pominąć, bowiem system automatycznie "wie", gdzie należy zapisać informacje. Zmienną środowiskową musisz utworzyć w dodatkowym katalogu "Datatypes", inaczej nie będzie działać. Dlatego też przed "Warp-PNG.prefs" umieściliśmy jego pełną nazwę.

Przypomnijmy, że jeśli chcesz odczytać informacje zapisane w zmiennej środowiskowej, należy użyć polecenia GETENV w podobny sposób, lecz podając tylko nazwę pliku:

STREFA AMIGI



getenv ENV:Datatypes/Warp-PNG.prefs

lub krócej:

getenv Datatypes/WarpPNG.prefs

Jak widać naszym elementem sterującym jest:

IMAGE_GAMMA 1500000

Dzięki takiemu zapisowi określasz argument oraz jego wartość - do wykonania przez datatyp. Działa to podobnie jak podczas wpisywania linii w oknie AmigaDOS, lecz podany parametr jest wywoływany za każdym razem, gdy skorzystasz z formatu PNG obsługiwanego przez zainstalowany, konkretny datatyp.

Zwróć uwagę, że dzieje się to niezależnie od użytej przeglądarki. Aby sprawdzić rezultat zmian, należy – po wykonaniu linii polecenia zawierającej SETENV – wyświetlić grafikę za pomocą dowolnej przeglądarki. Pamiętaj, że zmienne środowiskowe są domyślnie odczytywane przez datatyp, dlatego nie musisz wykonywać żadnych innych czynności.

Słowo "IMAGE_GAMMA" jest tylko przykładem pokazującym jak należy zmieniać ustawienia, ale od razu poznałeś pierwszą funkcję datatypu PNG. Powoduje ona zmianę algorytmy tak zwanej korekcji "gamma", o którym wspominaliśmy wcześniej. W praktyce parametr ten wpływa na jasność wyświetlanego obrazu i zwykle podajemy go w formie ułamka dziesiętnego, ze znakiem kropki "." jako separator, na przykład:

0.55

Program wymaga jednak pomnożenia wartości przez milion, co nie zmienia wywoływanej funkcji. Nasz ułamek musisz więc zapisać w następującej formie:

550000

Nie jest to typowy sposób obsługi, lecz łatwo można się go nauczyć. Wystarczy przyjąć, iż liczby wprowadzamy w "milionach" i sprawdzić efekt działania kilku skrajnych wartości. Spróbuj na przykład ustawić argument "IMAGE_GAMMA" na:

2.40

czyli po obliczeniu iloczynu:

2400000

Z pewnością zauważysz różnicę w wyświetlanym obrazie. Najbardziej odpowiednie wartości należy ustalić samodzielnie metodą prób i błędów.

Korekcja "gamma" to tylko jedna z wielu funkcji dostępnych w ramach pakietu "Warp PNG". Argument "IMAGE_GA-MMA" zmienia jasność grafiki, ale jego skuteczność nie jest zbyt duża. Dzieje się tak dlatego, iż pliki w formacie PNG mogą mieć zapisane własne parametry i zgodnie nimi ustawiana jest paleta barw. W konsekwencji część kolorystyki obrazu może stać się niewidoczna lub pokazana będzie nieprawidłowo.

Dużo lepszym rozwiązaniem jest zmiana współczynnika przypisanego do monitora. Dzięki temu jasność może być dopasowana do otoczenia wyświetlacza, na przykład, gdy Twój komputer stoi w ciemnym lub jasnym pomieszczeniu. Ponadto w takiej sytuacji sposób wyświetlania grafiki będzie dostosowywany proporcjonalnie do możliwości Twojego monitora, a nie ustawiany sztywno dla każdego pliku, niezależnie od zawartości.

Aby uzyskać ten efekt musisz skorzystać z argumentu o podobnej nazwie "SCREEN_GAMMA". Musisz go użyć w dokładnie tak samo jak wcześniej, czyli na przykład wpisując linię:

setenv Datatypes/WarpPNG.prefs SCREEN_GAMMA 2200000

Wartość ta jest domyślna i odpowiada ułamkowi dziesiętnemu:

2.2

Trzymaj się ogólnych uwag związanych z użyciem współczynnika "gamma", a z pewnością w niedługim czasie ustalisz parametry najbardziej odpowiadające wyświetlaczowi, który używasz. Korekcję "gamma" możesz również całkowicie wyłączyć. Jest to przydatne przede wszystkim wtedy, gdy chcesz obejrzeć obraz bez żadnej obróbki ze strony przeglądarki.

Standardowo funkcja ta jest aktywna i aby ją wyłączyć należy zastosować kolejny argument "GAMMA_CORREC-TION" w następujący sposób:

setenv Datatypes/WarpPNG.prefs GAMMA_CORRECTION OFF

Teraz program nie będzie brał pod uwagę żadnego z parametrów zawierających w nazwie słowo "GAMMA". Jeżeli będziesz chciał włączyć korekcję ponownie, zamiast "OFF" - wpisz "ON", czyli:

setenv Datatypes/WarpPNG.prefs GAMMA_CORRECTION ON

Datatyp nie udostępnia w tym miejscu bardziej szczegółowych ustawień, lecz to nie koniec jego możliwości.

Nastę- pna opcja to możliwość zmiany mechanizmu roztrząsania, czyli "ditheringu". Funkcja ta ma zastosowanie przede wszystkim w trybach wyświetlania o małej ilości dostępnych kolorów, ale jej efekty mogą być widoczne również na ekranach 15- i 16-bitowych. Wszystko zależy od rodzaju przeglą-



R,

danej grafiki. Nie należy również zapominać, że karta graficzna pozwala korzystać z trybów 256-kolorowych (8bitowych), które zajmują mniej pamięci. W związku z tym, w niektórych sytuacjach, "dithering" może okazać się jedynym wyjściem, aby wyświetlić obraz w rozsądnej jakości.

Pierwsza możliwość to określenie rodzaju trybu wyświetlania, dla którego efekt roztrząsania ma być aktywny. Należy tu użyć argumentu o nazwie "DITHER_DEPTH" podając minimalną głębokość ekranu dla działania funkcji. Oznacza to, że wpisujemy ilość "bitów", na przykład:

DITHER_DEPTH 8

spowoduje, że "dithering" będzie używany dla trybów wyświetlania o ilości kolorów powyżej 8-bitów, a dla tych z mniejszą ilością barw będzie nieaktywny. Możesz również określić sposób działania mechanizmu, co ma związek nie tylko z jakością grafiki jaką uzyskasz, ale także szybkością pracy przeglądarki. W tym przypadku skorzystaj z argumentu "DITHER_QUAL-ITY", który może przyjmować trzy wartości:

 POOR - najprostszy sposób dostosowania kolorów obrazu, jest też najszybszy, ale w zasadzie nie wykorzystuje techniki roztrząsania; dlatego w większości przypadków rezultat nie jest zadowalający,

- **GOOD** - domyślna metoda "ditheringu", wolniejsza, ale uzyskuje zdecydowanie lepszą jakość obrazu,

- **BEST** - opiera się na tym samym mechanizmie co poprzednia, lecz stara się wyświetlić obraz w najlepszej jakości; w związku z tym metoda ta jest również najwolniejsza.

Warto zwrócić uwagę, że w praktyce pomiędzy opcją "POOR" a pozostałymi,

różnice są zauważalne zawsze. Dwie ostatnie udostępniają podobną jakość grafiki, a ich użycie zależy od stopnia szczegółowości obrazu, zastosowanej rozdzielczości ekranu oraz rozmiaru pliku na dysku.

Możliwe jest również ustawianie sytuacji, w których roztrząsanie będzie działać w inny sposób, niezależny od poprzednich funkcji. Musisz tylko skorzystać z następnego argumentu o nazwie "DITHER OVERRIDE". Pozwala on sterować zachowaniem datatypów zależnie od programu, z którego zostaną wywołane. Innymi słowy możesz określać, czy ustawienia zawarte w zmiennej środowiskowej będą stosowane, czy mają być zastąpione przez inne wartości.

I tak, jeżeli nie chcesz, aby jakikolwiek inny program zmienił ustalone działanie funkcji roztrząsania – skorzystaj z opcji "NOTHING", czyli razem:

DITHER_OVERRIDE NOTHING

Drugą możliwością jest użycie standardowych preferencji związanych z systemowym datatypem "picture". Tak się będzie działo, gdy użyjesz argumentu tak jak poniżej:

DITHER_OVERRIDE DEFAULTS

Zwróć uwagę, że w tym przypadku wartości "ditheringu" zostaną zmienione, jeśli tylko program, z poziomu którego wyświetlasz grafikę, określi swoje własne ustawienia. Jest to ważne, bo nie zawsze wiemy jak wewnętrznie działa przeglądarka, ponadto ze składników systemu nie musi korzystać koniecznie tylko oprogramowanie służące do wyświetlania obrazów.

Przykładem jest pakiet "Super View", za pomocą którego można także wykonywać obróbkę i konwersję grafiki, a ustawienia datatypu PNG mają bezpośredni wpływ na jakość uzyskiwanych plików wynikowych.

Może zdarzyć się też potrzeba przyjęcia określonych początkowych ustawień roztrzasania, które później będą możliwe do zmiany przez program wywołujący. Taką możliwość daie następna wartość "APPS". Należy ją stosować w analogiczny sposób i nie ma ona dodatkowych ustawień. Ostatnią funkcją jest używanie zawsze własnych "preferencji" zewnętrznych programów. Wtedy wartości przyjęte w zmiennej środowiskowej nie będą miały żadnego znaczenia, bowiem system zawsze będzie korzystał z "obcych" ustawień. Nie jest to rozwiązanie zbyt bezpieczne, bo nie każdy program domyślnie określa sposób "ditheringu".

W tym przypadku pozbywasz się niejako kontroli nad tą funkcją, przynajmniej w tych sytuacjach, gdy będziesz korzystał z mniej rozbudowanego oprogramowania albo takiego, które nie przewiduje zmiany mechanizmu roztrząsania. Jednak czasem może się to przydać, na przykład podczas obróbki grafiki, kiedy ważne jest wczytywanie plików niepoddanych modyfikacjom. Jeżeli masz taką potrzebę, użyj wartości "EVERYTHING" w ten sam sposób, co poprzednio:

DITHER_OVERRIDE EVERYTHING

Wszystkie funkcje razem dają spore możliwości kontroli nad "ditheringiem" już na etapie obsługi składników systemowych, czyli datatypów. Pamiętaj, że możesz, a nawet powinieneś stosować je razem, inaczej włączysz tylko część dostępnych opcji.

Gdy używasz trybów wyświetlania 256kolorowych (czyli 8-bitowych), możesz wpływać na sposób dobierania tak zwanych "pisaków" systemowych w procesie dostosowywania kolorów. "Warp PNG" posiada dwie opcje związane z kontrolą pisaków, dzięki

STREFA AMIGI

którym możesz decydować jaka ilość barw będzie wykorzystywana do wyświetlania grafiki. Podobnie jak w przypadku roztrząsania, masz do dyspozycji ustawienia związane z jakością obrazu. Należy tu skorzystać z argumentu nazwanego "PENS_QUALITY", który posiada następujące opcje:

- **POOR** - używana będzie mała ilość pisaków, kosztem jakości obrazu,

 GOOD - domyślny sposób,
 który polega na użyciu takiej ilości barw, która gwarantuje prawidłowy wyświetlenie wszystkich podstawowych kolorów; pisaki nie są w tym przypadku w żaden sposób blokowane dla przeglądarki, dlatego mogą wystąpić przekłamania grafiki,

- **BEST** - pozwala wykorzystać jak największą ilość "pisaków", dzięki czemu uzyskuje najlepszą możliwą jakość; metoda ta nie bierze pod uwagę innych elementów graficznych, które mogą być w tym samym czasie widoczne na ekranie, dlatego podczas przeglądania różnych plików kolorystyka może być nieprawidłowa dla niektórych obrazów.

Sposób używania pisaków może być określany w różny sposób nie tylko w ramach Workbencha, lecz także innych programów.

Dlatego do dyspozycji masz kolejny argument "PENS_OVERRIDE", który przyjmuje te same wartości i funkcje jak omawiany wcześniej "DITHER_OVER-RIDE". Analogicznie możesz tu korzystać ze słów:

- NOTHING
- DEFAULTS
- APPS
- EVERYTHING

Wywołują one analogiczne rezultaty, lecz każdy z nich dotyczy "pisaków", a nie efektu roztrząsania. Podczas korzystania z formatu PNG masz możliwość kontroli tak zwanego kanału "alfa". Służy do tego argument "ALPHA_MODE", który umożliwia zmianę ustawień związanych z przezroczystością grafiki. Pierwsza opcja to wywołanie go wraz ze słowem "USE", czyli razem:

ALPHA_MODE USE

dzięki czemu kanał "alfa" jest łączony z kolorem tła i jest to standardowe zachowanie w całym systemie operacyjnym. Możesz z tej funkcji zrezygnować podając inną wartość argumentu, mianowicie:

ALPHA_MODE DISCARD

Teraz kanał "alfa" nie będzie w ogóle brany pod uwagę, co może być przydatne w sytuacji, gdy grafika zawiera obraz całkowicie przezroczysty. Niektóre programy mogą bowiem wyświetlać taki plik jako pusty obraz. W rezultacie obu opcji uzyskasz grafikę 24-bitową, zamiast 32-bitowej (czyli 24bity oraz kanał "alfa").

Możesz to zmienić przy użyciu kolejnej wartości "KEEP", która spowoduje utworzenie pliku 32-bitowego za pomocą systemowego datatypu o nazwie "picture".

Zwróć uwagę, że nie każda wersja obsługuje tryb "RBGA", który jest w tym przypadku wymagany. Z przezroczystością związany jest też argument "BACKGROUND", który nie ma żadnych dodatkowych ustawień. Działa tylko wtedy, gdy zastosujesz następujący argument:

ALPHA_MODE USE

i pozwala zmienić barwę tła. Jako wartość należy mu podać kolor w formacie szesnastkowym zwanym też heksadecymalnym. Na przykład:

BACKGROUND FF0000

lub

BACKGROUND 0xFF0000

spowoduje ustawienie koloru na czerwony. Tak się dzieje, bowiem dwa pierwsze znaki oznaczają wartość czerwonej składowej koloru, który chcesz uzyskać. Dwa kolejne odpowiadają za barwę zieloną, a ostatnie – za niebieską. Wszystko zawiera się w zwykłym schemacie kolorystycznym typu RGB.

Zwróć uwagę, że powyżej podane linie są równorzędne. Z przedrostka "0x" możesz zrezygnować, choć czasem jest przydatny, na przykład przy pisaniu własnych skryptów ARexxa.

To wszystkie ustawienia pakietu "Warp PNG", które mają bezpośredni wpływ na sposób i jakość wyświetlanego obrazu. Dla Amigi powstało bardzo wiele oprogramowania umożliwiającego zwiększenie możliwości systemu operacyjnego.

Pamiętaj, że każdy produkt będzie miał inne funkcje i własne "preferencje". Niektóre mogą nie posiadać rozbudowanych ustawień lub też posiadać specyficzne wymagania, nie tylko odnośnie zainstalowanych plików na dysku, ale także sposobu użycia. Jeżeli nie jesteś pewien jak korzystać z pobranych datatypów, w pierwszym rzędzie zawsze przeczytaj dołączoną dokumentację.

Obsługa formatu PNG to tylko przykład wykorzystania ogromnych możliwości systemu operacyjnego Amigi. Posiada on budowę modułową, dzięki czemu bez problemu można dodawać funkcje rozpoznawania nowych formatów nawet w starszych programach. Muszą one tylko korzystać z funkcji systemowych, co jest standardem w programach przeznaczonych dla Workbencha 2.0 lub nowszego.

Adam Zalepa

R

Kompresja na PowerPC

Dla procesorów PowerPC nie powstało tak wiele programów kompresujących dane jak dla serii Motorola 68000, niemniej za ich pomoca możesz pakować i. rozpakowywać większość popularnych formatów. Niektóre pozycje zachowują pełna funkcjonalność starszych wersji i należy je obsługiwać tak samo, inne sa bardziej ograniczone i w związku z tym posiadają zmienione nazwy funkcji.

Nie ma ścisłej reguły, bowiem programy powstawały w różnym okresie czasu i opierają się na odmiennych rozwiązaniach. Ważne więc, aby nauczyć się rozpoznawać sposób obsługi.

LHA

W przypadku podstawowego dla Amigi programu kompresującego dane, wersja dla procesorów PowerPC wygląda praktycznie identycznie jak dla serii Motorola 68000. Wersję działającą w środowisku pakietu "WarpOS" znajdziesz na Aminecie, w katalogu "util/arc". Należy pobrać plik o nazwie "Iha_wos.lha", a następnie rozpakować go w zwykły sposób, czyli za pomocą linii polecenia:

lha x lha_wos.lha RAM:

aby zawartość została zapisana w "Ram Dysku". Następnie odczytaj katalog docelowy i włącz tryb wyświetlania wszystkich plików. W oknie katalogu na Workbenchu znajdziesz ikonę podpisaną jako "lha wos". Jest to główny plik archiwizera LHA, odpowiadający funkcjonalnie zwykłej wersji. Należy go skopiować w miejsce, gdzie bedzie dostępny dla wszystkich programów, a więc najlepiej do systemowego katalogu "C". W tym celu najedź wskaźnikiem na ikonę "lha_wos" i naciśnij dwa razy lewy klawisz myszki.

Gdy pojawi się okno "Wykonaj polecenie" wpisz w nim:

copy lha_wos C:

Możesz także od razu zmienić nazwę polecenia, dzięki czemu będziesz mógł korzystać z wersji PowerPC dokładnie tak samo jak z przeznaczonej dla procesorów 68000.

Aby tak się stało zamiast powyższej linii wpisz:

copy lha_wos C:LhA

Pamiętaj jednak, że jeżeli wcześniej miałeś zainstalowaną inną wersję, zostanie ona automatycznie zastąpiona. Dlatego najlepiej wykonać kopię poprzedniego pliku lub zmienić nazwę na wygodniejszą, ale jednocześnie różniącą się od standardowego LHA. Na przykład:

copy lha_WOS C:Lha2

Wszystko zależy od tego, czy chcesz zachować poprzednie wydanie archiwizera, czy też planujesz korzystać wyłącznie z edycji przeznaczonej dla procesorów PowerPC. Pozostawiam to Twojej decyzji, bowiem nie ma to wpływu na funkcjonowanie obu poleceń. Oba pliki mogą być zainstalowane jednocześnie na dysku, ważne tylko, aby miały nazwy, których później nie pomylisz.

Wywołanie nowego LHA w oknie "Shell" nie różni się od tego, co należy zrobić przy używaniu programów dla procesorów serii 68k. Poza innym nagłówkiem informującym o wersji programu, zachowane są wszystkie poprzednie możliwości.

Mechanizmy archiwizera są zgodne z wersjami dla procesorów Motorola 68000, jak również pozwala on na wykonywanie czynności kompresji i dekompresji - bez ograniczeń. Również zastosowane metody pakowania plików są takie same, poza tym, że oczywiście całość będzie pracować szybciej. Zanim jednak zaczniesz na stałe korzystać z nowej wersji, proponuję

STREFX XMIGI



sprawdzić kilka przykładowych plików na okoliczność występowania błędów. Jeżeli opcja testowania i rozpakowania będzie działać bez problemów, możesz zrezygnować ze starszego wydania i używać go tylko w sytuacjach awaryjnych.

Przypomnijmy dwie podstawowe opcje, czyli sprawdzanie poprawności archiwum oraz dekompresja danych. Wywołujemy je za pomocą funkcji "t" (ang. test) oraz "x" (ang. extract), na przykład:

lha t Worek:dane3/pliki.lha

lub

lha x Worek:dane3/pliki.lha

Jak widać, należy robić to tak samo jak w przypadku "zwykłej" wersji programu, choć pamiętaj, że nie jest to regułą w całym oprogramowaniu dla procesorów PowerPC.

UNLZX

Sprawa wygląda nieco inaczej w przypadku archiwizera LZX. Stanowi on odpowiednik LHA, lecz pozwala na uzyskanie bardziej wydajnej i często szybszej kompresji. Posiada analogiczne funkcje, jednak wersja dla PowerPC jest bardziej ograniczona. Pozwala jedynie na rozpakowanie danych oraz wyświetlenie zawartości archiwum.

Program "UnLZX" znajdziesz oczywiście znowu na Aminecie, w tym samym katalogu co poprzednio, czyli "util/arc". Tym razem pobierz plik "PPCunlzx.lha" i rozpakuj go wpisując linię:

lha x PPCunlzx.lha RAM:

Teraz odczytaj "Ram Dysk" na Workbenchu i włącz funkcję wyświetlania wszystkich pozycji w katalogu. Zobaczysz ikonę podpisaną "PPCunlzx". Wykonaj na niej standardowy "dwuklik", aby wyświetlić kolejne ikony. Jedną z nich będzie plik "UnLZX.elf". Musisz go skopiować do systemowego katalogu "C", tak samo jak w przypadku LHA.

Ponownie wskazana jest zmiana nazwy, aby można było łatwiej korzystać ze wszystkich funkcji. Najedź więc wskaźnikiem na ikonę "PPCunlzx" i naciśnij dwa razy lewy klawisz myszki. W oknie "Wykonaj polecenie" wpisz linię:

copy UnLZX.elf C:UnLZX

i naciśnij ENTER. Zamiast "C:UnLZX", możesz także wpisać nazwę bardziej wyróżniającą się, abyś nie zapomniał, że masz do czynienia z edycją dla procesorów PowerPC. Powyższe rozwiązanie jest jednak najbardziej naturalne i wygodne, bowiem z nowego programu będziesz mógł korzystać tak samo, jak byłaby to wersja dla serii Motorola 68000 – poza oczywiście możliwością pakowania plików.

Zwróć także uwagę, że tym razem archiwizer działa pod kontrolą pakietu "PowerUP", a więc wymaga obecności biblioteki systemowej o nazwie "powerpc". Dodatkową informacją jest fakt, iż potrzebna jest wersja 46 lub nowsza. Możesz to sprawdzić za pomocą systemowego polecenia VERSION, w oknie "Shell" wpisując:

version ppc.library

Gdy zainstalujesz odpowiednie pliki, możesz przejść do właściwej pracy, a więc praktycznego korzystania z programu "UnLZX". Po wpisaniu jego nazwy w oknie AmigaDOS wyświetlony zostanie krótki komunikat mówiący o dwóch podstawowych funkcjach.

Zwróć uwagę, że trzeba ich używać w inny sposób niż wcześniej, mianowicie symbole poprzedzać znakiem myślnika.

Zwykle do tej pory oznaczało to dodatkową opcję, teraz w ten sposób wywołasz główne funkcje archiwizera.

Domyślną operacją jest dekompresja, dlatego jeżeli chcesz tylko rozpakować archiwum wystarczy wpisać linię zawierającą nazwę pliku, na przykład:

unlzx Worek:dane5/grafika.lzx

Spowoduje to rozpakowanie archiwum do katalogu bieżącego w oknie "Shell". Możesz też podać inne miejsce, w którym mają zostać zapisane pliki. Właściwy symbol należy dodać na końcu, czyli:

unlzx Worek:dane5/grafika.lzx Worek:

W tym przypadku pliki znajdujące się wewnątrz archiwum zostaną umieszczone w głównym katalogu dysku o nazwie "Worek:". Gdy chcesz mieć pewność, że uruchomisz funkcję dekompresji możesz także dodać symbol "-x" - zaraz po nazwie polecenia. Razem daje to linię w następującej postaci:

unlzx -x Worek:dane5/grafika.lzx Worek:

Działanie obu wersji jest identyczne, dlatego – o ile nie zależy Ci na maksymalnej czytelności – możesz pominąć znak "-x". Możesz również wyświetlić zawartość archiwum bez rozpakowania danych. W tym celu poprzedni symbol zastąp nowym: "-v" i usuń nazwę katalogu docelowego. Przykładowo:

unlzx -v RAM:moje_pliki.lzx

spowoduje wypisanie plików znajdujących się w archiwum "moje_pliki.lzx" umieszczonym w "Ram Dysku".

Możliwości wykorzystania procesorów PowerPC jest dużo więcej, na razie jednak zapoznaj się z podstawowymi

STREF# #MIGI



programami, których starsze odpowiedniki z pewnością masz zainstalowane w systemie i korzystasz z nich na co dzień. Dzięki temu będziesz mógł zaobserwować różnice w obsłudze tych samych plików oraz szybkości działania.

W następnych numerach naszego magazynu opowiem o składnikach ingerujących w pliki Workbencha, co jest dużo trudniejsze do instalacji, bo wymaga współdziałania kilku programów w ściśle określonych wersjach, w przeciwnym razie spowodujesz awarię całego systemu. Mimo wielu obaw użytkowników, system operacyjny Amigi radzi sobie bardzo dobrze z obsługą dwóch procesorów, nawet jeśli mają one pracować w wielozadaniowości. *Adam Zalepa*



Karta turbo CyberStorm z procesorem PowerPC 604e/233 MHz zamontowana w Amidze 4000.



Fraktale i inne figury

Jednym z najciekawszych tematów z dziedziny programowania grafiki są fraktale. Można je używać do symulowania wzrostu roślin, generowania realistycznych krajobrazów czy po prostu do prezentacji danych matematycznych w interesującej formie. Program, który chcę przedstawić pozawala wejść w świat fraktali na komputerze PC Jr.

Termin fraktal został wymyślony przez Benoita Mandelbrota, pioniera w badaniach oznaczania krzywych lub powierzchni mających wymiar ułamkowy. Pojęcie wymiaru ułamkowego można zilustrować następująco: prosta jest jednowymiarowa, ma tylko długość.

Jednak jeśli krzywa jest nieskończenie długa i zakrzywia się w taki sposób, aby całkowicie wypełnić obszar płaszczyzny, można ją uznać za dwuwymiarową. Natomiast krzywa częściowo wypełniająca obszar ma wymiar ułamkowy.

Wiele typów fraktali jest samo-podobnych, co oznacza, że wszystkie części fraktali przypominają się wzajemnie. Samo-podobieństwo występuje wtedy, gdy całość jest rozszerzeniem podstawowego elementu konstrukcyjnego. W języku fraktali ten podstawowy element konstrukcyjny nazywany jest generatorem. Generator w programach towarzyszących składa się z szeregu połączonych segmentów linii. Krzywe wykresów są wynikiem uruchomienia generatora, a następnie wielokrotnego zastępowania każdego segmentu linii całą procedurą generatora, wykonaną zgodnie z określoną regułą. Teoretycznie cykle te będą trwać w nieskończoność, ale w praktyce ogranicza je rozdzielczość ekranu.

Program numer 1 pokazuje to, co Mandelbrot nazywa "krzywymi smoczymi". Pokazuje krok po kroku sposób wypełniania krzywej fraktalnej. Generator składa się z dwóch odcinków linii o jednakowej długości tworzących kąt prosty.

Podczas każdego cyklu operacja jest wykonywane dla każdego segmentu znajdującego się na przemiennych bokach, czyli na lewo od pierwszego segmentu, na prawo od drugiego segmentu, i tak dalej. Program napisany jest w języku Basic, więc rysowania działania wystarczająco wolno, aby umożliwić obserwację rozwoju krzywej.

Program prosi o wprowadzenie parzystej liczby cykli. Ze względu na ograniczoną wydajność i rozdzielczość ekranu, kreślone są tylko cykle należące do liczb parzystych. Po zakończeniu działania, naciśnięcie dowolnego klawisza powoduje czyszczenie ekranu i powrót do początku. Polecam zacząć od dwóch cykli, potem czterech, sześciu itd. Aby całkowicie wypełnić "smoka", potrzeba czternastu cykli, ale ponieważ na wolniejszych maszynach może to wymagać nawet dwóch godzin pracy, prawdopodobnie po kilkunastu cyklach użytkownik będzie chciała zrezygnować. Aby zobaczyć kompletnego smoka wystarczy uruchomić Program numer 2, który wyrysuje go w ok. 30 sekund.

Nie jest wcale oczywiste, jak działa program, dlatego przedstawiam krótkie objaśnienie. Zmienna NC oznacza liczbę cykli, C jest numerem cyklu, natomiast SN jest tablicą liczb segmentów indeksowanych według numeru cyklu. Zmienna L to długość segmentu, D jest kierunkiem segmentu, ponumerowanym zgodnie z ruchem wskazówek zegara, a X i Y są współrzędnymi ekranu o wysokiej rozdzielczości.

W liniach 100-140 uzyskujemy liczbę cykli od użytkownika. Linia 150 Oblicza długość segmentu, a linia 160 Ustawia współrzędne początkowe. Z kolei,

Wiersze od 180 do 220 powodują odnalezienie kierunku segmentu w ostatnim cyklu. Linie 230-260 zwiększają lub zmniejszają wartość X lub Y o długość segmentu, w zależności od kierunku. Wiersze od 270 do 290 kreślą segment i aktualizują numer bieżącego segmentu dla każdego cyklu. Linie 300Program numer 1

MH 90 DIM SN(14): KEY OFF MN 100 CLS:SCREEN 0 PI 110 PRINT "ENTER AN EVEN NO. 0 F CYCLES (2 TO 14)" DD 120 INPUT"OR ENTER A ZERO TO QUIT: ";NC EH 130 IF NC = 0 THEN KEY ON: END MO 140 IF NC MOD 2 = 1 OR NC < 2 OR NC > 14 THEN 100 DL 150 L = 128: FOR C=2 TO NC STEP 2:L-L/2:NEXTAE 160 X = 192:Y=133:CLS:SCREEN 2: PSET <X,Y),1 DL 170 FOR C = 0 TO NC:SN(C)=0:NEXT K KJ 180 D=0:FOR C=1 TO NC: IF SN(C-1)=SN(C) THEN D=D-1:GOTO 200 GC 190 D=D+1 BP 200 IF D=-1 THEN D=7 MI 210 IF D=8 THEN D=0 NC 220 NEXT OA 230 IF D=0 THEN X=X+L+L:GOTO 270 EH 240 IF D=2 THEN Y=Y+L:GOTO 270 GA 250 IF D=4 THEN X=X-L-L: GOTO 270 GM 260 Y=Y-L PL 270 LINE -(X,Y), 1:SN(NC) -SN(NC) + 1 QP 280 FOR C=NC TO 1 STEP -1: IF SN(C) <>2 THEN 300 OH 290 SN(C) = 0: SN(C-1) = SN(C-1) + 1: NEXTGA 300 IF SN(0)=0 THEN 180 MM 310 IF INKEY\$="" THEN 310 AC 320 GOTO 100

320 zawierają warunek mówiący, iż jeśli numer segmentu dla zerowego cyklu nadal wynosi zero, należy wykonać operację dla następnego segmentu, w przeciwnym razie kończymy działanie programu.

Program numer 2 kreśli ponad 8000 różnych smoków. Osiąga to poprzez losowe ustalanie, po której stronie pierwszego segmentu zostanie zastąpiony generator dla wszystkich cykli. Generator jest zawsze podstawiony po lewej stronie pierwszego segmentu w pier-Poza wszym cyklu. czynnikiem losowym, program ten wykorzystuje tę samą logikę co Program numer 1. Główna część jest napisana w języku maszynowym, aby skrócić czas wymawykreślenia całkowicie gany do wypełnionego smoka.

Wszystkie smoki są wykreślane po 14 cyklach. Wszystkie zajmują dokładnie

ten sam obszar, który jest równy połowie kwadratu odległości między narysowanymi pierwszymi i ostatnimi punktami. Wszystkie smoki zaczynają się i kończą w tych samych punktach. Po zakończeniu działania naciśnij Spację, aby wykreślić innego smoka, lub naciśnij klawisz Q, aby zakończyć program.

Program numer 3 przedstawia figury, które Mandelbrot określa jako "płatki śniegu". Segmenty są ponumerowane od zera do sześciu, zaczynając od prawej. Listing jest w zasadzie taki sam jak Program numer 1.

Zmienne NC, C, SN, D, X i Y reprezentują te same wartości, z wyjątkiem tego, że D jest numerowana w kierunku przeciwnym do ruchu wskazówek zegara. Dla każdego segmentu dołączona tabela podaje wartość RD (kierunek względny), LN (współczynnik długości) i SD (flagi wskazujące, po której stronie segmentu ma zostać umieszczony generator).

Linia 20 odczytuje wartości SD i RD, a następnie oblicza wartości LN. Linie od 30 do 50 obliczają współczynniki delta x i delta y dla każdego kierunku.

Wiersze 60-100 pozwalają uzyskać liczbę cykli od użytkownika, natomiast linia 120 ustawia współrzędne początkowe. Linia 130 Ustawia numery segmentów dla wszystkich cykli dla pierwszego segmentu.

Wiersze od 140 do 170 znajdują kierunek segmentu w ostatnim cyklu. Linie 180-190 obliczają współrzędne końca segmentu, rysują segment i aktualizują numery segmentów dla każdego cyklu. Wiersze od 200 do 220 mają to samo działanie, co linie 300-320, cały czas w Programie numer 1.

Podobnie jak w przypadku programu 1, naciśnięcie dowolnego klawisza po zakończeniu rysowania powoduje czyszczenie ekranu i wyświetlenie kolejnego pytania o liczbę cykli.

Mam nadzieję, że zaprezentowane programy zachęcą Was do zainteresowania się fascynującym światem fraktali. Struktury o budowie fraktalnej są powszechnie spotykane w przyrodzie, tak więc nasze rozważania nie są czystą matematyką. Nie bójmy się też eksperymentować z programami - spróbujmy na przykład zmodyfikować kształt generatora w Programie numer 3 lub zaprojektować własny generator. Jest to łatwiejsze niż może się wydawać, a daje bardzo dużo satysfakcji i można uzyskać nietypowe efekty na ekranie.

> Opracował: Leszek Rosiński
PP 100 DEF SEG: CLEAR, &H3FF0:N=&H4000 LL 110 READ A\$: IF A\$="/" THEN 130 LI 120 POKE N,VAL("&H" + A\$) :N=N + 1: GOTO 110 OC 130 N = &H440F: FOR K=1 TO 15:POKE N,0:N = N + 1: NEXT NH 140 POKE &H4425, 0 IF 150 N=&H4000:CALL N:POKE &H4425, 1 EA 160 A\$ = INKEY\$:IF A\$ = "" THEN 160 PI 170 IF A\$=" " THEN 150 ID 180 IF A\$<>"Q" AND A\$ <>"q" THEN 160 IH 190 SCREEN 0: CLS: KEY ON: END EI 1000 DATA 1E, 0E, IF, B8, 05, 00, CD, 10, 80, 3E DN 1010 DATA 25, 44, 00, 75, 0B, B4, 00, CD, 1A, 89 NF 1020 DATA 16, 23, 44, EB, 31, 90, BE, 02, 00, B9 OE 1030 DATA 08, 00, A1, 23, 44, 33, D2, A9, 02, JE 1040 DATA 74, 02, B2, 01, A9, 04, 00, 74, 02, 00 02, B6 NG 1050 DATA 01, 32, D6, D0, EA, D1, D8, E2, E8, BJ 1060 DATA 23, 44, 24, 01, 88, 84, 0F, 44, 46, A3 83 FH 1070 DATA FE, 0F, 75, D3, B8, 00, 06, 33, C9, BA LI 1080 DATA 4F, 18, 32, FF, CD, 10, B9, 0F, 00, NJ 1090 DATA F6, C6, 84, 00, 44, 00, 46, E2, FB, 33 C7 44, 60, 00, C7, 00, 44, 01, 0C, 8B, 0E, 1E, 44, 01, 0C, 8B, 0E, 1E, 44, 44, JD 1100 DATA 06, 1E, HF 1110 DATA 00, B8, 60, 00, C7, 06, 20, 84 8B, 16 DC 1120 DATA 20, 44, CD, 10, C6, 06, 22, 44, 00, B9 LO 1130 DATA 0E, 00, 33, FF, BE, 01, 00, 8A, A5, KC 1140 DATA 44, 80, FC, 00, 75, 18, FE, 06, 22, 0F 44 CL 1150 DATA 8A, 85, 75, 00, 44, 3A, 84, 00, 44, 20 LG 1160 DATA FE, OE, 22, 44, FE, OE, 22, 44, EB, 16 OD 1170 DATA FE, OE, 22, 44, 8A, 85, 00, 44, 3A, 84 HK 1180 DATA 00, 44, 75, 08, FE, 06, 22, 44, FE, 06 DC 1190 DATA 22, 44, 80, 3E, 22, 44, FF, 75, 07, C₆ HN 1200 DATA 06, 22, 44, 07, EB, 0C, 80, 3E, 22, 44 NN 1210 DATA 08, 75, 05, C6, 06, 22, 44, 00, 47, 46 FE 1220 DATA E2, AB, EB, 02, EB, 9A, 80, 3E, 22, 44 CH 1230 DATA 00, 75, 06, FF, 06, 1E, 44, EB, 1E, KP 1240 DATA 3E, 22, 44, 02, 75, 06, FF, 06, 20, 80 02, 75, 06, FF, 06, 20, 44 IG 1250 DATA EB, 11, CO 1260 DATA OE, 1E, 80, 3E, 22, 44, 04, 75, 06, FF 44, EB, 04, FF, 0E, 20, 44, B8 ID 1270 DATA 01, OC, 8B, OE, IE, 44, 8B, 16, 20, 44 LL 1280 DATA CD, 10, FE, 06, 0E, 44, BF, 0D, 00, BE KL 1290 DATA OE, 00, B9, OE, 00, 80, BC, 00, 44, 02 PI 1300 DATA 75, 0D, C6, 84, 00, 44, 00, FE, 85, 00 FM 1310 DATA 44, 4F, 4E, E2, EC, 80, 3E, 00, 44, 00 OK 1320 DATA 75, 02, EB, 9C, 1F, CB,/ Program numer 3 DE 10 DIM DX(11), DY(11):KEY OFF MC 20 FOR N = 0 TO 6: READ SD(N), RD(N) : LN(N)=1!/3!:NEXT:LN(2) = SQR(LN(1))LC 30 A = 0: FOR D = 6 TO 11:DX(D) = COS(A):DY(D) = SIN(A)JN 40 A = A + .52359879#:NEXT CK 50 FOR D = 0 TO 5:DX(D) = - DX(D + 6):DY(D) = - DY (D + 6): NEXT:X1 = 5 34:Y1 = 147: TL = 324OG 60 CLS: SCREEN 0 AB 70 PRINT "ENTER NUMBER OF CYCLES (1 - 4)" GK 80 INPUT "OR ENTER A ZERO TO QUIT: ";NC JA 90 IF NC = 0 THEN END DA 100 IF NC > 4 THEN 60 OP 110 CLS:SCREEN 2 CL 120 X = 534: Y = 147: TL = 324: PSET(X,Y), 1 CD 130 FOR C = 0 TO NC:SN(C) = 0:NEXTMI 140 D = 0: L = TL: NS = 0: FOR C = 1 TO NC: I = SN(C): L = L + LN(I) : J = SN(C-1) : NS = NS + SD(J) : IF NS MOD 2 = 1 THEN D=D+12-RD(I) : GOTO160 GE 150 D=D+RD(I) EG 160 D = D MOD 12 OL 170 NEXT BN 180 X = X + 1.33*L*DX(D) : Y = Y-.5*L*DY(D) : LINE - (X, Y), 1: SN(NC) = SN(NC) + 1: FOR C = NC TO 1 STEP -I: IF SN(C) <> 7 THEN 200 OG 190 SN(C) = 0: SN(C-1) = SN(C-1) + 1: NEXT AH 200 IF SN(0) = 0 THEN 140 KE 210 IF INKEY\$="" THEN 210

Program numer 2

Pliki wsadowe w IBM Basic

Temat plików wsadowych był popularny w w latach '90-tych, ale to nie znaczy, że dzisiaj nie możemy wykorzystać tego sposobu kontroli działania komputera. Trzeba jednak powiedzieć, że plik wsadowy MS-DOS-u jest dość ograniczony funcjonalnie. Dużo lepsze rezultaty możemy osiągnąć wywołując DOS z poziomu IBM Basica, dzięki czemu wywołamy wiele dodatkowych funkcji.

Tak jak na innych komputerach, także w przypadku peceta możemy za-

File Edit Search Options

SET SOUND=C:NDRIVERSNSB16

eECHO OFF

oszczędzić wiele czasu używając automatycznych poleceń systemu MS-DOS. Jednak nie jest to najlepszy sposób działania ze względu na liczne ograniczenia.

Możliwe jest automatyczne wykonywanie sekwencję poleceń, ale dość surowy interfejs DOS-u zapewnia tylko trzy wariacje jednej prostej instrukcji IF i nie ma żadnego praktycznego sposobu na manipulowanie zmiennymi typu ciąg lub wykonywanie bardziej skomplikowanych działań arytmetycznych.

Bardzo trudno jest napisać plik wsadowy, który tworzy rozbudowane ekrany i wykonuje logiczne, pozwalając na wprowadzanie danych przez użytkownika. Pewnym problemem jest też przechwytywanie błędów.

Bardziej elastyczną techniką jest wywoływanie poleceń MS-DOS lub plików wsadowych z poziomu programu Basica. To uwalnia programistę od ograniczeń wspomnianych w poprzednim akapicie. DOS można wywołać z poziomu IBM Basica za pomocą instrukcji SHELL, która z pewnością wielu osobom będzie się kojarzyć z oknem komunikacyjnym w systemie Amigi, czyli AmigaDOS.

Polecenie to wymaga wersji systemu 2.1 lub wyższej. Instrukcja SHELL znajduje się również w PCjr (Cartridge BA-SIC), ale nie działa zbyt dobrze z powodu konfliktów pamięci. Polecenie SHELL może przekazać ciąg bezpośrednio lub zmienną łańcuchową.

Podczas korzystania z tej techniki należy pamiętać o dwóch ważnych kwestiach. Najpierw upewnij się, że Twój system ma wystarczającą ilość pamięci. Chociaż DOS, interpreter Basica i program mogą być ładowane nawet do maszyny z zaledwie 64 kilobajtami pamięci RAM, nie pozostanie Ci dużego obszaru roboczego.

Dlatego zalecane jest posiadanie przynajmniej 92 kilobajtów wolnej pamięci. Oczywiście wywołanie pliku wsad-

9FI PF49FFV-4556 12 h1 H2 L226 10
SET MIDI=SYNTH:1 MAP:E
C:NDRIVERSNSB16NDIAGNOSE /S
C:NDRIVERSNSB16NMIXERSET /P /Q
C:NDOSNSMARTDRU.EXE /X
PROMPT \$p\$g
PATH C:NDOS
SET TEMP=C:NTMP
LOADHIGH=C:\DOS\MSCDEX.EXE /D:MSCD001
DOSKEY
÷
MS-DOS Editor <f1=help> Press ALT to activate menus</f1=help>

AUTOEXEC.BAT

Przykład najbardziej typowego pliku wsadowego.

owego wymaga większej ilości, w praktyce kompletne środowisko pracy zajmie początkowo minimum 80 kilobajtów RAM.

Z dzisiejszego punktu widzenia jest to ilość śmiesznie mała, nawet jak na standardy komputera pracującego tylko pod kontrolą systemu MS-DOS, ale pamiętajmy, że kupując retro peceta nie zawsze będziesz miał możliwość łatwej rozbudowy pamięci. Wszystko zależy od rodzaju kości pamięci, ewentualnych konfliktów, które w tego typu sprzęcie zdarzają się często.

Po drugie, pamiętaj, aby nie tworzyć sekwencji, która jest ciągła lub rekurencyjna. Na przykład wynik będzie nieprzewidywalny, jeśli program w Basicu wywoła plik wsadowy, który z kolei wywoła z powrotem Basic. Sekwencje tego rodzaju mogą spowodować awarię systemu, którą można naprawić tylko poprzez reset komputera, a więc z skrajnym przypadku można stracić wprowadzony do pamięci listing, czego oczywiście nie chcemy.

Kolejny drobny problem polega na tym, że DOS przewija 25 linii na ekranie, podczas gdy BASIC obsługuje tylko 24 linie ze względu na wyświetlanie klawiszy funkcyjnych w ostatniej linii. Ponadto, Basic i DOS utrzymują niezależny wskaźnik lokalizacji określającej pozycję kursora. Różnice te mogą spowodować, że instrukcje Basica takie jak PRINT zastąpią komunikaty wypisane przez DOS.

Aby uniknąć tego problemu, uruchamiaj program w Basicu zawsze przy użyciu polecenia KEY OFF, co spowoduje wyłączenie wyświetlania klawiszy funkcyjnych.

Następnie użyj polecenia CLS (czyszczenie ekranu) za każdym razem, gdy MS-DOS wraca do języka Basic. Dodatkowo można stosować poniższą konstrukcję:

LOCATE 24,1 Print Print

aby upewnić się, że zarówno DOS, jak i Basic rozpocznie przewijanie zawartości ekranu zawsze w dolnej części.

Osobiście niedawno ponownie stałem się posiadaczem zestawu z procesorem MMX 200 MHz, dlatego o moich kolejnych doświadczeniach z DOS-em będę donosił w kolejnych numerach naszego magazynu.

> Opracował: Leszek Rosiński

Więcej na temat programowania w Basicu na komputerach PC oraz o plikach wsadowych można przeczytać na poniższych stronach:

https://robhagemans.github.io/pcbasic/doc/1.2/

http://www.chebucto.ns.ca/~ak621/DOS/BatBasic.html

http://gobruen.com/progs/
dos_batch/dos_batch.html



PCjr i program w jednej linii

Czy zastanawialiście się kiedyś, jak duży program można umieścić w jednej linii w IBM Basic? Chcę pokazać przykład, który być może zachęci Was do spróbowania swoich sił w tym języku. Mój przykład działa na komputerze PCjr lub dowolnym PC z kolorową kartą graficzną.

Jak wiadomo, programy nie muszą być długie, aby mogły być uznane za przydatne. Program zaprezentowany w ramce może być z powodzeniem wykorzystany do zabawy, na przykład ucząc dzieci alfabetu i układu klawiatury komputera. Dorośli mogą czerpać inne korzyści, zdobywając większą wiedzę na temat wyświetlania różnych znaków na ekranie.

Nie zapomnij zapisać programu. Kiedy go uruchomisz, pozornie nic się nie dzieje. Teraz naciśnij dowolny klawisz zawierający literę. Duży znak pasujący do wciśniętego klawisza pojawi się z dołu ekranu i zostanie wycentrowana. Dalej naciśnij inny klawisz. Ta sama procedura będzie powtarzana do czasu, aż naciśniesz kombinację Ctrl-Break lub Fn-Break na komputerze PCjr.

Program powiększa wszystkie litery (wielkie i małe), symbole i znaki diakrytyczne. Jeśli naciśniesz Ctrl razem z literą lub liczbą, zobaczysz jeden ze znaków graficznych dostępnych w standardowym zestawie IBM-a. Nawet klawisze funkcyjne dają swoje własne wyniki, tak więc naciśnij na przykład klawisz F1.

Pomimo niewielkich rozmiarów program zawiera kilka technik, które mogą okazać się przydatne także podczas pisania innych programów. Zaczynamy od ustawienia segmentu na adres \$FFA6. Jest to 14 bajtów przed mapą definicji znaku w pamięci ROM, czyli tylko do odczytu. Program odczytuje Twoje naciśnięcie klawisza i konwertuje go na odpowiedni kod ASCII. Wartość mapy przechowywana jest w zmiennej A i inicjalizowana jako ciąg A\$.

Zmienna M jest równa jeden, jeśli bit jest ustawiony, w przeciwnym razie wynosi zero. Zmienna W równa się 32 (jest to kod ASCII dla Spacji), jeśli bit nie jest ustawiony lub 219 (jest to kod ASCII dla wypełnionej Spacji). Trzy puste pola lub trzy stałe bloki są dodawane na początku zmiennej A\$. Następnie program przesuwa wszystkie bity w bieżącym bajcie, odejmując wartość prawego i dzieląc przez liczbę 2. Ten krok jest powtarzany dla każdego bitu bieżącego bajtu.

Następnym krokiem jest wyświetlenie A\$ poprzedzone 28 znakami Spacjami. Odbywa się to dwukrotnie, aby podwoić wysokość widocznego znaku. Dalej program przechodzi do etapu drugiego i powtarza proces dla każdego bajtu mapy, która definiuje znak. Po przetworzeniu ostatniego bajta mapy wypisujemy dwie puste linie i wracamy do kroku pierwszego, aby poczekać na kolejne naciśnięcie klawisza.

Bardzo zaawansowani programiści mogą zauważyć, że kod może być jeszcze bardziej zwarty. Niektóre linie można dodatkowo połączyć, ale spowoduje to, że nawiasy będą miały kilka poziomów zagnieżdżenia, co utrudni zrozumienie programu.

Jak widać, nawet bardzo krótkie programy mogą mieć ciekawe działanie, a ich analiza może przynieść sporo niespodzianek.

> Opracował: Leszek Rosiński

1 KEY OFF:DEF SEG=&HFFA6:L\$= INPUT\$(1) : N=ASC(L\$) : CLS:LOCATE 24,1,0:FOR L=0 TO 7 : A=PEEK (N*8+L+14) : A\$="" : FOR J=0 TO 7:M=A AND 1 : W=32+M*187: A *=CHR\$ (W)+CHR\$ (W)+CHR\$ (W)+A\$: A=(A-M)/2:NEXT J:PRINT SP C(28) A\$: PRINT SPC(28) A\$: NEXT L : PRINT : PRINT : GOTO 1

Quadscreen

Większość naszych komputerów w trybie tekstowym oferuje 40- lub 80-kolumny. Jest to ogólnie zadowalające dla większości aplikacji do przetwarzania tekstu. Jednak w przypadku korzystania z arkusza kalkulacyjnego, który nie pracuje w sposób graficzny, jest to dość uciążliwe i może bvć mylace po wprowadzeniu większej ilości danych. Quadscreen to produkt zaprojektowany w celu wyeliminowania tego problemu. Powiększa ekran komputera IBM i daje mu możliwość generowania wyświetlacza o wymiarach 160 na 64, czyli wystarczająco dużego dla większości arkuszy kalkulacyjnych.

Quadscreen jest pakietem sprzętowoprogramowym, który zawiera 17-calowy monochromatyczny monitor wysokiej rozdzielczości, pozwalający wyświetlać o obraz bez migotania, czyli efektu Interlace. Rozdzielczość wynosi 968 punktów poziomych oraz 512 linii pionowych, tak więc na ekranie można jednocześnie umieścić maksymalnie ponad 10 tys. znaków.

Karta graficzna Quadscreen zawiera 128-kilobajtową pamięć RAM mieszczącą wiersze o pojemności 1 KB każdy, z których każde kolejne 512 może być jednocześnie wyświetlane na ekranie. Do zestawu dołączone są dwa dyski instalacyjne oraz kod źródłowy. Najtrudniejszą częścią instalacji jest ustawienie przełączników DIP na karcie graficznej. Niestety nie udało mi się zdobyć dokumentacji, tak więc jeśli ktokolwiek posiada oryginalną instrukcję - proszę o kontakt na mój adres mailowy. Karta działa jak każda inna karta graficzna, czyli po prawidłowym ustawieniu przełączników, komputer startuje jak zwykle i możemy przejść do instalacji oprogramowania.

Jest to bardzo proste - wystarczy uruchomić DOS i z dysku instalacyjnego uruchomić program QSINIT. Standardowy sterownik umożliwia cztery tryby wyświetlania. Na szczęście tym razem wszystkie ważne informacje pojawiają się na ekranie. Podczas pracy w trybie Quadscreen określamy liczbę znaków w linii i typ przewijania oraz wybieramy tryb pełno-ekranowy lub podzielony tryb Nastepnie oprogramowanie pracy. samoczynnie wybiera czcionkę, która najlepiej pasuje do wybranych opcji. Chodzi o to, aby na ekranie zmieściła się określona ilość tekstu.

Mamy też narzędzie FONTEDIT, które pomaga w tworzeniu i edycji nowych czcionek. Można je następnie zastosować do oprogramowania sterownika, aby zastąpić oryginalne kroje. Znak są edytowane na siatce za pomocą klawiszy funkcyjnych i klawiatury numerycznej. FONTTEDIT jest równie łatwy w użyciu, jak inne generatory znaków, ale dobrym rozwiązaniem byłoby dodanie sterowania myszką. Szkoda też, że nie możemy tworzyć biblioteki znaków. Trzeba przyznać, że duży obszar wyświetlania robi wrażenie, jak na produkty z pierwszej połowy lat '80-tych. Jednak nie wszystkie programy będą działać na wyświetlaczu Quadscreen, bowiem sterowniki ekranowe znajdują się w pamięci systemowej. W związku z tym część oprogramowania może nadpisywać sterowniki i komputer przestanie wyświetlać obraz.

Podsumowując, jest to unikalny sposób na używanie programów biurowych z tamtych lat, choć problemy z kompatybilnością sprawiają, że można uruchamiać tylko część oprogramowania, jak na przykład VisiCalc. Z pewnością miało to wpływ na małą popularność tego rozwiązania. Mimo wszystko warto dostrzec, że w okresie sprzed standaryzacji rozwiązań na rynku PC, różni producenci szukali własnego sposobu na korzystanie z komputera i uzyskanie większych możliwości także w nietypowy sposób.

