

SPARTA DOS X

Dyskowy System Operacyjny

SpartaDOS X v. 4.40

Suplement do wydanej przez PZ KAREN instrukcji użytkownika SpartaDOS X uwzględniający zmiany w stosunku do wersji 4.20.

© 2006-2008 DLT Ltd.

Rozdział 2: Wprowadzenie do SpartaDOS X

Ustawienie czasu i daty

Domyślnym formatem daty przyjmowanym przez DOS od wersji 4.40 jest format europejski (dd-mm-yy). Użytkownik może go zmienić na tradycyjny amerykański (mm-dd-yy). W związku z tym polecenie **DATE** interpretera poleceń wyświetla odpowiedź, w jakiej kolejności należy wpisać poszczególne liczby składające się na datę:

```
Current date is 10-09-06  
Enter new (DD-MM-YY):
```

Uruchamianie programów

Poprzedzenie nazwy programu znakiem # ma taki sam efekt, jak użycie polecenia X, np. wpisanie:

```
#DISKRK
```

będzie miało takie samo działanie jak wpisanie:

```
X DISKRK
```

Konieczność uruchomienia „przez X” występuje w przypadku większości programów napisanych dla AtariDOS. Użycie polecenia X (lub #) jest na ogół właściwym sposobem usunięcia kłopotu, jeśli przy próbach uruchamiania programów uporczywie pojawia się błąd „Memory conflict”. Gdy to nie pomaga, potrzebne jest zapewne przekonfigurowanie DOS-u (patrz rozdział 8 – Konfiguracja systemu).

Posiadaczy szybkich twardej dysków ucieszy wiadomość, iż możliwe jest obecnie takie skonfigurowanie DOS-u, że wykona on tę

czynność (tj. odłączenie modułu biblioteki w celu zwolnienia miejsca na wczytywany program) automatycznie.

Odpowiedniki DOS 2

M – RUN AT ADDRESS

Polecenie zewnętrzne RUN

Rozdział 3: Przegląd SpartaDOS X

Identyfikatory urządzeń

Od wersji 4.40 DOS obsługuje do piętnastu napędów podłączonych jednocześnie. Są one ponumerowane od 1 do 15. Dyski o numerach od 1 do 9 mają, tak samo jak w poprzednich wersjach SpartaDOS, identyfikatory 1: ... 9: lub A: ... I: z poziomu interpretera poleceń, lub D1: ... D9: oraz DA: ... DI: z poziomu BASIC-a.

Dyski o numerach od 10 do 15 oznaczane są wyłącznie literami, tj. J: ... O: z poziomu interpretera poleceń lub DJ: ... DO: z poziomu BASIC-a.

W systemie, oprócz DSK:, CAR:, PRN: i COM: istnieje teraz dodatkowe urządzenie NUL:. Przyjmuje ono dowolną ilość zapisywanych danych (nie zachowując ich nigdzie), przy odczycie natomiast zachowuje się na jeden z trzech sposobów:

NUL: lub NUL1: – zwraca błąd 136 (EOF).

NUL2: – zwraca nieskończony ciąg zer.

NUL3: – zwraca nieskończony ciąg liczb losowych.

Zgodność formatu dyskietek

SpartaDOS X 4.40 używa formatu dyskietki lekko zmodyfikowanego w stosunku do poprzednich wersji SpartaDOS. Format ten jest zgodny wstecz z formatem stosowanym przez SpartaDOS X 4.x oraz SpartaDOS 2.x i 3.x, tak więc nie powinno być żadnych kłopotów z wymianą danych pomiędzy tymi wersjami SpartaDOS.

Nie należy natomiast spod SpartaDOS X 4.40 dokonywać jakichkolwiek operacji zapisu (w tym kasowania plików) na dyskietkach SpartaDOS 1.1, jeśli mają być jeszcze kiedyś używane ze SpartaDOS 1.1. Zaleca się w takich przypadkach przekopiowanie plików przeznaczonych do obróbki na dysk w formacie SpartaDOS X.

SpartaDOS X 4.40 jest w stanie obsługiwać dyski z sektorami o wielkości 512 bajtów. Pliki zapisane na takim dysku nie są dostępne dla żadnej innej wersji SpartaDOS, wliczając w to poprzednie wersje SpartaDOS X, oraz, o ile nam wiadomo, w ogóle wszystkie DOS-y działające na Atari.

Rozdział 4: Command Processor – polecenia

APPEND

Zadanie: dołącza podaną ścieżkę dostępu do \$PATH

Składnia: APPEND *ścieżka*

Typ: zewnętrzny – na urządzeniu CAR:

Program dołącza podaną ścieżkę dostępu do aktualnej zawartości zmiennej \$PATH.

ARC

Błąd polegający na tym, że ARC wywołany z parametrem XH mógł wyświetlić komunikat i zatrzymać się czekając na odpowiedź użytkownika przy wyłączonym ekranie, został poprawiony. Dodatkowo na pytanie, czy wyciąganymi z archiwum danymi należy nadpisać istniejący już plik dyskowy o tej samej nazwie, można teraz odpowiedzieć, oprócz „No” lub „Yes”, również „All”.

BASIC

Polecenie BASIC uruchamia wewnętrzny moduł Atari BASIC-a również na komputerach wyposażonych w rozszerzenie pamięci do 1088 kB. W poprzednich wersjach SpartaDOS X było to niemożliwe.

Zmienna środowiskowa BASIC nie ma teraz początkowo żadnej wartości, o ile nie zainstalujemy sterownika RAMDISK.SYS. Jeśli w momencie załadowania tego sterownika zmienna BASIC nie ma wartości nadanej przez użytkownika (poleceniem SET w pliku CONFIG.SYS), sterownik ten ustawia ją tak, żeby wskazywała plik BAS.SAV znajdujący się w ramdysku.

BLOAD

Zadanie: wczytuje wskazany plik pod wskazany adres w pamięci.

Składnia: BLOAD fname.ext [\$]adres

Typ: zewnętrzny – na urządzeniu CAR:

Plik o podanej nazwie wczytywany jest jako blok danych pod podany adres, po czym sterowanie wraca z powrotem do interpretera poleceń DOS-u.

Program nie sprawdza, czy plik mieści się w pamięci albo nie nakłada się na obszary ważne dla systemu operacyjnego – zakłada się, że użytkownik korzystający z tego polecenia wie, co robi.

Poleceniem służącym do wczytywania wykonywalnych plików

binarnych jest LOAD.

CHKDSK

W poprzednich wersjach SpartaDOS X CHKDSK było poleceniem wewnętrznym interpretera poleceń, teraz natomiast jest poleceniem zewnętrznym, realizowanym przez program CHKDSK.COM umieszczony na urządzeniu CAR: (CAR:CHKDSK.COM).

Uruchomienie CHKDSK z parametrem /X powoduje odczytanie i wyświetlenie poszerzonej informacji o parametrach dysku. Gdy podano parametr /V, dodatkowo odczytywana i analizowana jest mapa bitowa dysku (VTOC), a uzyskana stąd informacja o ilości wolnego miejsca jest porównywana z ilością wolnego miejsca wykazaną w sektorze nr 1. Pozwala to na szybkie sprawdzenie, czy na dysku znajdują się tzw. „utracone sektory”. Opcja /V działa tylko dla dysków w formacie SpartaDOS.

COLD

W wersji DOS-u przeznaczonej do zaprogramowania w module typu Maxflash użycie polecenia COLD /C nie ma sensu.

COMP

Zadanie: porównuje wskazane pliki.

Składnia: COMP [d:][path]fname1.ext [d:][path]fname2.ext

Typ: zewnętrzny – na urządzeniu CAR:

Program porównuje wskazane pliki i wyświetla listę różnic między nimi. Jeśli program podczas porównywania działa nierówno – tj. raz szybciej, raz wolniej, coraz wolniej wraz z oddalaniem się od początku obu plików, przy czym głowica stacji dysków bardzo intensywnie przemieszcza się pomiędzy ścieżkami – może to świadczyć, że liczba buforów zadeklarowanych dla SPARTA.SYS jest zbyt mała.

COPY

COPY jest poleceniem wewnętrznym interpretera poleceń, które jednak tylko uruchamia program CAR:COPY.COM. Istnieje możliwość zmiany tego i zastąpienia CAR:COPY.COM zewnętrznym programem. Służy do tego zmienna środowiskowa COPY, np. wpisanie:

```
SET COPY=C:>SYS>CP.COM
```

spowoduje, że interpreter poleceń uruchomi wskazany program w

momencie wywołania polecenia COPY, i przekaże mu wszystkie, podane przez użytkownika parametry.

DATE

SpartaDOS X wyświetla teraz datę i czas w formacie europejskim (dd-mm-yy) lub amerykańskim (mm-dd-yy) w zależności od konfiguracji nadanej przez użytkownika. Dotyczy to również polecenia DATE. Do zmiany formatu daty służy zmienna środowiskowa DAYTIME. Wpisanie:

```
SET DAYTIME=1
```

wybiera format amerykański, a

```
SET DAYTIME=2
```

format europejski. Brak ustawienia zmiennej lub ustawienie DAYTIME=0 wybiera format domyślny, którym w obecnej wersji DOS-u jest format europejski.

DELTREE

Zadanie: rekursywne usuwanie całych drzew podlatalogów.

Składnia: DELTREE [d:][path]dirname

Typ: zewnętrzny – na urządzeniu CAR:

Program po uruchomieniu jeszcze pyta o potwierdzenie, a po odpowiedzi twierdzącej rekursywnie usuwa wskazany katalog wraz z całą zawartością, sukcesywnie raportując postępy.

Przerwanie tej czynności komunikatem *Can't delete directory* oznacza, że na dysku znajduje się wpis katalogowy pliku, który został otwarty do zapisu, ale wskutek jakiegoś błędu (np. zawieszenia się komputera) pozostał nigdy nie zamknięty. Taki wpis jest niewidoczny w katalogu i nieusuwalny normalną metodą, ale powoduje, że katalog nawet po skasowaniu wszystkich znajdujących się w nim plików wciąż nie jest pusty – a więc nie może zostać usunięty. Taka sytuacja jest nieprawidłowa i świadczy o naruszeniu struktury systemu plików. Do automatycznego naprawiania tego typu błędów służy program CleanUp z pakietu SpartaDOS Toolkit.

DF

Zadanie: wyświetla zbiorczą informację o ilości wolnego miejsca na wszystkich dyskach.

Składnia: DF

Typ: zewnętrzny – na urządzeniu CAR:

Program wyświetla listę wszystkich dysków podając dla każdego z osobna następujące informacje: identyfikator dysku, całkowitą pojemność w sektorach, liczbę wolnych sektorów, liczbę wolnych kilobajtów, procent zajętości, nazwę dysku. Na samym dole listy wyświetlane jest podsumowanie.

DIR i DIRS

Poprzednie wersje SpartaDOS wyświetlały w katalogu tylko sześć ostatnich cyfr długości pliku, mimo że pliki o ośmiocyfrowej wielkości są jak najbardziej możliwe do stworzenia i są poprawnie – poza tym szczególnie – obsługiwane. Wszelako niemożność poprawnej oceny wielkości dużych plików jest uciążliwa.

Dla rozwiązania tej trudności w SpartaDOS X 4.40 wprowadzono zmianę polegającą na tym, że jeśli rozmiar pliku przekracza 999999 bajtów, jego wielkość w katalogu wyświetlana jest w kilobajtach, a dla zaznaczenia tego faktu na końcu liczby dodawana jest literka 'k'. Może to wyglądać na przykład tak:

```
Volume:      TEST  
Directory:  MAIN  
  
BACKUP    TAR    6988k  10-09-06  15:55  
51483    FREE    SECTORS
```

Krótki format katalogu zwracany przez DIRS wyświetla rozszerzenia katalogów (zamiast **DIR**). Jako oznaczenie podkatalogu stosowany jest dwukropek, podobnie jak w MyDOS.

Dodatkowy parametr /A powoduje wyświetlenie atrybutów w liście plików. Stosuje się go najczęściej w połączeniu ze znakiem "+", np. polecenie **DIR + /A** pokaże wszystkie pliki z ich atrybutami.

DPOKE

Polecenie zewnętrzne DPOKE jest analogiczne do POKE, z tym tylko, że umieszcza w pamięci pod wskazanym adresem wartości dwubajtowe w zwykłej konwencji (młodszy bajt pod adresem *adres*, starszy pod *adres+1*).

DUMP

Dodano parametr /A powodujący zamianę znaków charakterystycznych dla ATASCII (semigrafiki, znaków w inverse video

itp.) na kropki. Pozwala to na wydrukowanie zawartości pliku na drukarce również wtedy, kiedy interfejs drukarki nie przeprowadza pełnej translacji ani nie drukuje w trybie graficznym.

ECHO

Zadanie: włączanie i wyłączenie „echa” poleceń CP.

Składnia: ECHO ON|OFF

Typ: wewnętrzny, polecenie COMMAND.COM.

ECHO OFF powoduje, że polecenia wpisywane przez użytkownika w linii komend COMMAND.COM, lub pobierane z pliku wsadowego, nie są widoczne na ekranie. ECHO ON przywraca normalną sytuację.

ED

Zadanie: edytor tekstowy.

Składnia: ED [d:][path][filename.ext]

Typ: zewnętrzny – na urządzeniu CAR:

Jest to przystosowana do współpracy ze SpartaDOS X, relokowalna wersja programu JBW Edit. Przeznaczeniem jest przede wszystkim edycja plików konfiguracyjnych SpartaDOS X, ale może być użyty do obróbki dowolnych plików tekstowych, byle niezbyt wielkich (praktyczna granica znajduje się gdzieś w kolicach 6-8 kB, mimo że bufor edytora jest znacznie większy).

Gdy w linii komend podano specyfikację pliku, program podejmie próbę jego załadowania. Dostępne są następujące komendy edycyjne:

Esc – opuszczenie funkcji (Cancel) lub wyjście z programu.

Ctrl/L – Load – załadowanie wskazanego pliku do bufora edycyjnego.

Ctrl/S – Save – zapis zawartości bufora do wskazanego pliku.

Ctrl/U – Up – podniesienie dolnego marginesu „okna” edycyjnego.

Ctrl/D – Down – opuszczenie dolnego marginesu „okna” edycyjnego.

Ctrl/V – Visible – wyświetlenie znaków końca linii.

Ctrl/B – Begin – przerwienie kursora na początek tekstu.

Ctrl/E – End – przerwienie kursora na koniec tekstu.

Ctrl/A – przerwienie kursora na początek linii.

Ctrl/Z – przerwienie kursora na koniec linii.

Ctrl/T – Tag – zaznaczenie bieżącej linii.

Ctrl/G – Go – przerwienie kursora do zaznaczonej linii.

Ctrl/Q – Quit – wyjście z trybu interpretacji komend, następna kombinacja klawiszy będzie przyjęta jako znak.

Shift/Ctrl/strzałka w górę – przeskok o stronę wstecz.

Shift/Ctrl/strzałka w dół – przeskok o stronę w przód.

Shift/Insert – wstawienie bieżącej linii przed zaznaczoną (zob. Ctrl/T), kursor przechodzi do następnej linii.

Shift/Ctrl/E – Erase – usunięcie całej zawartości bufora edycyjnego.

ELSE

Patrz rozdział 5, „Pliki wsadowe”.

EXIT

Patrz rozdział 5, „Pliki wsadowe”.

FI

Patrz rozdział 5, „Pliki wsadowe”.

FORMAT

Formatter został dostosowany do obsługi większych dysków (o pojemności do 32 MB i z sektorami 512-bajtowymi) oraz do większej liczby widocznych w systemie napędów. Napędy wybiera się w funkcji U (jak *Unit*) wciskając klawisze liter z zakresu od A (napęd nr 1) do O (napęd nr 15). Natomiast wybór gęstości (*Density*) oferuje, oprócz trzech tradycyjnych, tj. *Single*, *Dual*, *Double*, czwartą gęstość *DD 512* – jest to gęstość podwójna z sektorami 512-bajtowymi. Jej wykorzystanie jest możliwe z niektórymi typami twardego dysku (KMK/JŻ/IDEa) oraz ze stacjami dysków TOMS – zarówno firmowymi, jak i mającymi rozszerzenie TOMS Turbo Drive lub TOMS Multi Drive.

Poprzednie wersje SpartaDOS formatują dyskietki w ten sposób, że ostatni sektor na dysku jest nieużywany i oznaczony jako zajęty. Funkcja *Optimize*, kiedy jest włączona, powoduje przypisanie ostatniego sektora dysku do obszaru danych, przez co na świeżo sformatowanej dyskietce mamy o jeden wolny sektor więcej niż kiedyś.

Istnieją wszelako bardzo stare twarde dyski, w których pierwszy sektor fizycznie istniejący ma numer 0, a nie 1, jak być powinno; jest on niedostępny, ale uwzględniany przez kontroler w zwracanej komputerowi informacji o całkowitej liczbie sektorów. Wadę tę mają np. dyski Supra Corp. i K-Products. Na takim dysku sektor ‘odzyskiwany’ przez funkcję ‘Optimize’ w rzeczywistości nie istnieje, dlatego posiadacz takiego urządzenia powinien wyłączyć optymalizację przed zapisaniem katalogu na dysku.

Uwaga: polecenie FORMAT bardzo intensywnie wykorzystuje obszar stosu 6502. W związku z tym niektóre rozszerzenia, w których procedury obsługi ładowane są na zwykle nieużywaną część stosu, przestaną działać. Problemy wystąpią np. w przypadku systemu turbo do stacji

dysków typu Top Drive 1050, czy jakiegokolwiek innego, gdzie program szybkiej transmisji ładowany jest na stos – z taką stacją transmisja będzie mogła być tylko normalna (tak samo zresztą, choć z innych względów, jak przy wcześniejszych wersjach SpartaDOS X).

Pozostałe systemy turbo, czyli TOMS Multi Drive (w trybie Ultra Speed, nie w trybie Top Drive), TOMS 710/720, CA-2001, Atari XF551, LDW 2000 Super, Indus GT, Happy 1050, US Doubler – będą działać jak dotychczas.

FSTRUCT

Zadanie: analiza struktury plików binarnych.

Składnia: FSTRUCT [d:][path]filename.ext

Typ: zewnętrzny – na urządzeniu CAR:

Program wypisuje na ekranie informację o strukturze pliku binarnego, którego nazwa podana została jako parametr.

FSYMBOL

Zadanie: konwersja wewnętrznego symbolu SpartaDOS X na adres.

Składnia: FSYMBOL nazwa_symbolu

Typ: zewnętrzny – na urządzeniu CAR:

Komenda podaje adres związany z podanym symbolem. Zobacz też SL.

IF

Patrz rozdział 5, „Pliki wsadowe”.

GOSUB i GOTO

Patrz rozdział 5, „Pliki wsadowe”.

MAN

Zadanie: przeglądarka plików tekstowych (głównie instrukcji do programów)

Składnia: MAN [filename]

Typ: zewnętrzny – na urządzeniu CAR:

MAN.COM jest prostą przeglądarką plików tekstowych przeznaczoną do szybkiego wywoływania na ekran instrukcji do komend lub programów o nazwach podanych jako parametr. Działanie to podobne

jest do komendy *man* (co jest skrótem od angielskiego słowa *manual* oznaczającego podręcznik lub instrukcję obsługi) znanej z systemu UNIX, i stąd taka a nie inna nazwa.

MAN wymaga zdefiniowania zmiennej środowiskowej \$MANPATH, która będzie zawierała listę katalogów do przeszukania przez program (podobnie jak zmienna \$PATH zawiera listę katalogów do przeszukania, kiedy użytkownik poda nazwę programu do uruchomienia). Separatorem poszczególnych katalogów, tak samo jak w przypadku \$PATH, jest średnik:

```
SET MANPATH=C:>MAN;D:>DOC
```

Teraz jeśli użytkownik wkopiuje do któregoś z tych katalogów pliki tekstowe z rozszerzeniami *.MAN, *.DOC lub *.TXT, podanie nazwy takiego pliku – ale bez rozszerzenia – spowoduje wyświetlenie go na ekranie.

Przykładowo, w archiwum HDSC.ARC znajduje się plik HDSC.DOC zawierający stosowne informacje na temat programu HDSC.EXE. Rozpakowawszy archiwum możemy plik wykonywalny umieścić w którymś z katalogów wskazywanych przez \$PATH, a plik *.DOC – w katalogu wskazywanym przez \$MANPATH. Od tego momentu, by poczytać instrukcję programu HDSC, nie musimy już pamiętać, gdzie jest oryginalne archiwum, wystarczy wydać komendę:

```
MAN HDSC
```

a tekst wyświetli się na ekranie.

W przypadku tekstów napisanych „jednym ciągiem”, gdzie pojedynczy wiersz nie mieści się na ekranie, MAN próbuje złamać tekst dostosowując go do aktualnej szerokości ekranu.

MAP

Zadanie: konfiguracja SIO.SYS.

Składnia: MAP [unit] [SIO|OS|NORMAL|OFF] [d:]
 lub MAP [filename.ext]

Typ: zewnętrzny – na urządzeniu CAR:

Polecenie „mapuje” podany identyfikator dysku na napęd o podanym numerze (*unit*). Wykorzystywana jest tutaj wewnętrzna tablica translacji SIO.SYS. Dodatkowe opcje dotyczą metody prowadzenia transmisji dla danego dysku:

NORMAL – standardowa obsługa transmisji, urządzenie PBI ma priorytet nad SIO

SIO – obsługa wyłącznie urządzenia SIO (pominięcie PBI)
OS – przekierowanie obsługi dysku do ROM-u systemu operacyjnego
OFF – brak obsługi dysku (lub obsługa przez inny sterownik).

Parametr „SIO” umożliwia dostęp do napędu SIO (np. stacji dysków), który został przesłonięty przez napęd PBI o tym samym numerze. Na przykład, polecenie **MAP 1 SIO D2:** tworzy logiczny dysk D2: odwołujący się do fizycznej stacji dyskietek nr 1 (dyskiem D1: może być partycja twardego dysku PBI).

Parametr „OS” jest stosowany wtedy, gdy mają być użyte procedury obsługi transmisji zawarte w systemie operacyjnym komputera. W ten sposób mogą zostać włączone specjalne tryby transmisji, np. Top Drive, TOMS Turbo, o ile system operacyjny potrafi je obsłużyć. Działanie w trybie “OS” nie jest możliwe, gdy DOS jest skonfigurowany do używania pamięci pod ROM-em systemu operacyjnego (USE OSRAM).

Druga składnia wywołania MAP umożliwia jednoczesną zmianę ustawień większej ilości napędów, przyjmując jako parametr nazwę pliku konfiguracyjnego. Jego zawartość powinny stanowić linie, z których każda odpowiada pierwszej składni polecenia MAP (od parametru “unit”).

Uwaga: MAP nie działa, jeśli zamiast SIO.SYS zainstalowany jest sterownik SIOOLD.SYS!

MDUMP

Zadanie: wyświetla pamięć w formie liczb szesnastkowych i znaków ATASCII.

Składnia: MDUMP [\$]adres

Typ: zewnętrzny – na urządzeniu CAR:

Program działa na pamięci podobnie jak DUMP na pliku. Przydatny do szybkiego sprawdzania zawartości pamięci.

MEM

Polecenie MEM jest teraz zewnętrzne, realizuje je program zawarty w urządzeniu CAR: (CAR:MEM.COM). MEM wyświetla też dodatkową informację o bieżącej konfiguracji pamięci systemowej, to jest nazwę trybu (NONE, OSRAM lub BANKED). Po podaniu przełącznika /X wyświetlana informacja jest nieco bardziej szczegółowa, między innymi podawany jest kod banku, w którym rezyduje DOS, co może być informacją istotną dla programistów i zaawansowanych użytkowników.

Informacja o liczbie dostępnych banków rozszerzenia pamięci może się różnić od tego, co na konkretnym komputerze podawała starsza wersja SpartaDOS X. Przyczyną jest fakt, że SpartaDOS 4.2x nie radził

sobie z poprawnym rozpoznaniem wielkości niektórych rozszerzeń, i tak np. 256k TOMS (12 banków) widziane było jako 192k (8 banków). W wersji 4.40 obsługa pamięci została poprawiona.

PROC

Patrz rozdział 5, „Pliki wsadowe”.

RENAME

Działanie komendy RENAME (oraz odpowiedniej funkcji kernela DOS-u) zostało udoskonalone tak, żeby nie można było nadać plikowi nazwy, którą ma już inny plik wpisany do tego samego katalogu.

RENDIR

Zadanie: zmiana nazw katalogów.

Składnia: RENDIR [d:][path]stara_nazwa nowa_nazwa

Typ: zewnętrzny – na urządzeniu CAR:

Program działa dla katalogów tak samo, jak komenda RENAME dla plików.

RETURN

Patrz rozdział 5, „Pliki wsadowe”.

RUN

Zadanie: uruchamia program od podanego adresu.

Składnia: RUN [\$]adres

Typ: zewnętrzny – na urządzeniu CAR:

Program wykonuje skok JMP pod podany adres. Nie sprawdza, czy ma to sens czy nie – zakłada się, że użytkownik korzystający z tego polecenia wie, co robi.

SETERRNO

Patrz rozdział 5, „Pliki wsadowe”.

SIOSET

Zadanie: konfiguracja przyspieszonej transmisji w SIO.SYS.

Składnia: SIOSET [d: [type [usindex]]]

Typ: zewnętrzny – na urządzeniu CAR:

Polecenie SIOSET służy do zaawansowanej konfiguracji trybów przyspieszonej transmisji, obsługiwanych przez sterownik SIO.SYS. W typowych przypadkach parametry transmisji zostają ustawione automatycznie przez sterownik SIO.SYS i nie ma potrzeby ich zmiany. W określonych sytuacjach (np. wymiana stacji podczas pracy komputera) mogą one zostać ustawione ręcznie.

Jeżeli w poleceniu nie zostaną podane żadne parametry, wówczas wyświetlana jest aktualna konfiguracja wszystkich napędów. Wartości parametru *type* mają następujące znaczenie:

RESET – skasowanie ustawień transmisji dla napędu, zostaną one ustalone przy pierwszym odwołaniu się do tego napędu przez system.

NORMAL – napęd nie obsługuje przyspieszonej transmisji.

XF – napęd pracuje w standardzie XF-551.

US – napęd obsługuje tryb UltraSpeed.

INDUS – napęd pracuje w standardzie Indus.

W przypadku UltraSpeed można za pomocą wartości *usindex* określić szybkość transmisji. Przykładowo, tzw. tryb 3xSIO (57,6 kbps) wymaga podania \$08.

Uwaga: SIOSET nie działa, jeśli zamiast SIO.SYS zainstalowany jest sterownik SIOOLD.SYS!

SL

Zadanie: wypisanie na ekran listy symboli SpartaDOS X.

Składnia: SL

Typ: zewnętrzny – na urządzeniu CAR:

Program wypisuje na ekranie listę wewnętrznych symboli systemowych SpartaDOS X.

SORTDIR

Zadanie: sortowanie katalogów według nazw, rozszerzeń, daty lub rozmiarów plików.

Składnia: SORTDIR [d:][path] [/NTSDX]

Typ: zewnętrzny – na urządzeniu CAR:

Program odczytuje wskazany katalog, sortuje go według podanych kryteriów, po czym zapisuje z powrotem w postaci posortowanej. Kryteria sortowania:

/N – według nazw plików

/T – według typu pliku (rozszerzenia nazwy)

/S – według wielkości plików

/D – według daty i czasu

/X – w porządku malejącym

Specyfikacja pliku nie musi być podana, SORTDIR przyjmuje wtedy bieżący katalog do obróbki, ale kryterium sortowania jest parametrem obowiązkowym. SORTDIR wywołany bez parametrów wyświetla krótką informację na swój temat i listę opcji.

Przy sortowaniu według nazwy, rozszerzenie stanowi kryterium drugie co do ważności. Przy sortowaniu według rozszerzenia, druga co do ważności jest nazwa pliku. Przy sortowaniu według rozmiaru, nazwa jest druga, a typ trzeci co do ważności. Cyfry mają w kolejności sortowania mniejszą wartość niż litery. Wszystko jest sortowane domyślnie w porządku rosnącym, podanie parametru /X odwraca ten kierunek.

SWAP

Polecenie SWAP nie będzie działać na napędy od J: do O:.

TD

Polecenie TD zostało poprawione (usunięto „problem roku 2000”). Format wyświetlania czasu zależy od aktualnej wartości zmiennej DAYTIME (zob. polecenie DATE). Litera „X” obrazuje dodatkowo stan Caps/Inverse video. Zobacz też opis procedury Z.SYS (rozdział ósmy).

UNERASE

Program realizujący komendę UNERASE w SpartaDOS X 4.20 ma znany od dość dawna błąd, który powoduje uszkodzenie danych bitmapy, jeśli odzyskiwany plik jest zawarty w więcej niż jednym sektorze, a seria bitów reprezentujących zajęte przezeń sektory przechodzi przez granicę sektorów bitmapy. Ten błąd poprawiono, test znajdujący się w pakiecie NNTOOLS Nelsona Nievesa wykonuje się teraz bezbłędnie.

VDEL

Zadanie: selektywne kasowanie w grupach plików.

Składnia: VDEL [d:][[path]filename.ext

Typ: zewnętrzny – na urządzeniu CAR:

Program znajduje pliki pasujące do podanego wzorca nazwy, wyświetla kolejno ich nazwy i za każdym razem pyta, czy skasować czy nie. Na końcu wyświetlane jest podsumowanie ilości skasowanych plików. Program można opuścić w każdej chwili wciskając **Esc**.

VDEL przydaje się w (rzadkich) razach, kiedy z grupy wielu plików trzeba wykasować część, dla której nie ma wspólnego i wyróżniającego spośród innych wzorca nazwy pliku nadającego się do podania jako maska dla polecenia **ERASE**. Pliki trzeba wtedy wykasowywać pojedynczo, VDEL pozwala przynajmniej częściowo zautomatyzować ten proces. Zadanie tego typu można też wykonać przy użyciu programu MENU.

XFCNF

Zadanie: wybieranie gęstości w stacji dysków.

Składnia: XFCNF [d:] [/12345]

Typ: zewnętrzny – na urządzeniu CAR:

Niektóre stacje dyskietek elastycznych mają kłopoty z automatycznym przełączaniem się pomiędzy gęstościami, zwłaszcza dotyczy to stacji Atari XF551 (stąd nazwa programu), ale nie tylko. XFCNF pozwala w momencie takiego „zacięcia” się stacji w jednej gęstości wymusić na niej przejście w inną – najczęściej odpowiednią dla tkwiącej w napędzie dyskietki.

Program uruchomiony bez parametrów wyświetla komunikaty, w których prosi o wybranie numeru napędu (od 1 do 4) oraz gęstości, w jaką ma ustawić stację. Po przesłaniu wybranej konfiguracji do napędu sprawdzane jest, czy stacja zaakceptowała nową gęstość rzeczywiście czy pozornie (niektóre stacje przyjmują bez zastrzeżeń rozkaz ustawienia się w gęstość, jakiej nie potrafią zrealizować, będzie to np. gęstość 360k dla Atari 1050 Top Drive, albo 720k dla Atari XF551 – kontroler stacji w takim przypadku wybiera na ogół coś najbardziej zbliżonego). Komunikat *Drive cannot do this density* oznacza stwierdzenie właśnie takiej pozornej akceptacji.

Pozostałe komunikaty:

Drive not configurable – napęd jest zwykłą stacją Atari 810 lub 1050 (bez przeróbek), zmiana gęstości jest niemożliwa.

Drive is not a floppy disk – programem próbowano zmienić gęstość ramdysku lub partycji twardego dysku.

Drive rejected the density – stacja odrzuciła żadaną konfigurację jako niemożliwą do realizacji.

XFCNF można obsługiwać z linii komend, bez konieczności wchodzenia do menu. Podanie samego identyfikatora napędu powoduje sprawdzenie, czy stacja jest napędem dysków elastycznych i czy jest konfigurowalna – odpowiedź *Drive is configurable* zawiadamia, że tak jest. Zmianę gęstości uzyskujemy podając ukośnik z cyfrą od 1 do 5, gdzie kolejne cyfry oznaczają gęstości jak poniżej:

/1 – gęstość pojedyncza (SSSD, 90k)

- /2 – gęstość średnia (SSED, 130k)
- /3 – gęstość podwójna (SSDD, 180k)
- /4 – gęstość podwójna dwustronna, 40 ścieżek (DSDD, 360k)
- /5 – gęstość podwójna dwustronna, 80 ścieżek (DSDD, 720k)

Inne komendy

Oprócz programów wymienionych powyżej, w urządzeniu CAR: są też dwa programy przeznaczone dla użytkowników interfejsów twardych dysków:

S2I – dla użytkowników SIO2IDE

MNT – dla użytkowników KMK/JŻ IDE

Opis użytkowania każdego z nich powinien znajdować się w dokumentacji odpowiedniej dla każdego z interfejsów.

Rozdział 5: Command Processor – szersze możliwości

Uruchamianie programów binarnych

Jeśli program binarny wymaga do pracy zwolnienia obszaru normalnie zajętego przez bibliotekę (adresy \$A000-\$BFFF), trzeba go uruchamiać poleceniem X. Ale poprzedzenie nazwy programu znakiem # ma taki sam efekt, jak użycie polecenia X, np. wpisanie:

```
#DISKRK
```

będzie miało takie samo działanie jak komenda:

```
X DISKRK
```

PLIKI POLECEŃ (WSADOWE)

Znajdująca się w pliku wsadowym linia tekstu, która zaczyna się od średnika (;), traktowana jest jako komentarz i pomijana w interpretacji.

Zawarcie w pliku poleceń komedy ECHO OFF blokuje wyświetlanie na ekranie odczytywanych z pliku wsadowego komend. ECHO ON odblokowuje je.

Począwszy od wersji 4.41 SpartaDOS-u uruchomienie pliku wsadowego powoduje automatyczne wyłączenie echa – w razie potrzeby trzeba je włączyć podając ECHO ON jako jedną z komend pliku wsadowego.

Wyrażenia warunkowe

Słowa kluczowe IF, ELSE i FI pozwalają na stosowanie prostych konstrukcji warunkowych. Ogólna składnia takowego jest następująca:

```
IF [NOT] EXISTS [atr] pathname|ERROR [n]|INKEY ['c'|n]
```

```
...  
FI
```

lub

```
IF [NOT] EXISTS [atr] pathname|ERROR [n]|INKEY ['c'|n]
```

```
...  
ELSE
```

```
...  
FI
```

Wyrażenia warunkowe mogą być zagnieżdżane jedno w drugim,

dozwolone jest do 255 poziomów zagnieżdżeń.

Warunek IF EXISTS

Operator EXISTS pozwala sprawdzić, czy podany plik lub katalog istnieje na dysku. Na przykład wyrażenie IF EXISTS FOO.BAR jest prawdziwe, gdy plik nazwany „FOO.BAR” istnieje w katalogu bieżącym. NOT powoduje logiczne zanegowanie wyniku, a więc IF NOT EXISTS FOO.BAR jest prawdziwe, gdy ten plik nie istnieje. Normalnie można w ten sposób sprawdzać tylko obecność zwykłych plików, wykrycie istnienia katalogu wymaga podania atrybutu w zwykły sposób: IF EXISTS +S FOO jest prawdziwe, gdy podkatalog o nazwie „FOO” istnieje w katalogu bieżącym.

Przykład:

```
IF EXISTS %1.ARC
  IF NOT EXISTS +S %1
    ECHO Creating dir %1
    MD %1
    ARC X %1 %1>
  ELSE
    ECHO %1 already exists
  FI
ELSE
  EXIT 170
FI
```

Po zapisaniu tego pliku pod nazwą, powiedzmy, X.BAT i umieszczeniu gdzieś wzdłuż \$PATH, wpisanie w linii poleceń:

```
-X PLIKI
```

spowoduje rozpakowanie archiwum PLIKI.ARC do automatycznie utworzonego w tym celu katalogu.

Warunek IF ERROR

Operator ERROR pozwala wykrywać błędy powstałe podczas wykonywania pliku poleceń. IF ERROR jest prawdziwe, jeśli wystąpił jakikolwiek błąd. Podobnie IF NOT ERROR jest prawdziwe, gdy nie było błędów. Można też podać konkretny numer błędu, np. IF ERROR 170 jest prawdziwe, jeśli ostatnim błędem, jaki wystąpił, był błąd nr 170 (File not found).

Słowo kluczowe ERROR używa zmiennej ERRNO, którą biblioteka systemowa tylko zapisuje, ale nigdy jej nie kasuje. Oznacza to, że znajdujący się tam kod błędu nie musiał zostać koniecznie wygenerowany przez polecenie ostatnio wykonywane w pliku wsadowym. Dlatego dobrze jest podać SETERRNO 0 przed wykonaniem komendy, której rezultat chcemy sprawdzić przez IF ERROR.

Warunek IF INKEY

Napotkanie tego operatora powoduje zatrzymanie wykonywania pliku wsadowego i oczekiwanie na naciśnięcie klawisza. Wyrażenie IF INKEY jest prawdziwe, gdy użytkownik nacisnął dowolny klawisz (oprócz Break – wtedy jest fałszywe). Można też sprawdzać naciśnięcie konkretnego klawisza, za wyrazem INKEY trzeba wtedy podać, jego kod ATASCII w postaci numerycznej (dziesiętnej lub szesnastkowej), albo odpowiadający mu znak ujęty w apostrofy. Na przykład, gdy chcemy zaprogramować oczekiwanie na klawisz „A” (o kodzie ATASCII 65), wyrażenie może mieć jedną z następujących postaci:

```
IF INKEY 65
IF INKEY $41
IF INKEY 'A'
```

Konstrukcja tego typu jest przydatna przy dokonywaniu prostych wyborów, jednak niezbyt się nadaje do skonstruowania czegoś bardziej skomplikowanego, np. menu z wieloma opcjami. Do tego służy, opisana dalej, komenda INKEY.

Operator porównania

Wykonanie pewnych fragmentów pliku wsadowego można uzależnić od wartości zmiennych środowiskowych. Służy do tego operator porównania, którym jest znak równości (=). Np. wyrażenie IF DAYTIME=2 jest prawdziwe, gdy zmienna środowiskowa DAYTIME zawiera tekst „2”. Oddzielnego operatora „jest różne” nie ma, warunek nierówności bada się poprzedzając równość operatorem negacji logicznej NOT, np. IF NOT DAYTIME=2 oznacza „jeśli DAYTIME jest różne od 2”.

Skoki GOTO

Komenda GOTO pozwala wykonać skok bezwarunkowy w obrębie pliku wsadowego. Składnia:

```
GOTO etykieta
```

Wykonanie tego przenosi wykonywanie pliku wsadowego do wiersza znajdującego się po tym, który zawiera definicję podanej etykiety. Przykładowa etykieta może być zdefiniowana jak następuje:

```
: ETYKIETA
```

Tak wygląda definicja. Podając nazwę etykiety jako argument do polecenia GOTO trzeba pominąć dwukropek.

Należy pamiętać, że komenda GOTO w poszukiwaniu podanej etykiety zawsze przeszukuje cały plik wsadowy od początku. Dlatego im dalej od początku pliku etykieta się znajduje, tym wolniej wykona się skok.

Komenda INKEY

Komenda INKEY zatrzymuje wykonywanie pliku wsadowego, czeka na naciśnięcie klawisza, a następnie przypisuje odpowiadającą mu literę wskazanej zmiennej środowiskowej. Wartość tę można stamtąd odczytać przy użyciu operatora porównania komendy IF. W ten sposób staje się możliwe skonstruowanie np. menu wielokrotnego wyboru:

```
:MENU
CLS
ECHO A. Opcja nr 1
ECHO B. Opcja nr 2
ECHO C. Koniec
INKEY KLAWISZ
IF KLAWISZ=A
    ECHO Wybrano opcje 1
FI
IF KLAWISZ=B
    ECHO Wybrano opcje 2
FI
IF KLAWISZ=C
    SET KLAWISZ
    EXIT
FI
PAUSE
GOTO MENU
```

Przeostrog: dane zapisywane są do obszaru zmiennych środowiskowych, który liczy tylko 256 bajtów. Żeby go nie przepełnić, dobrą praktyką jest usuwanie zmiennych tworzonych przez plik wsadowy, tak jak to pokazano w przykładzie (piąta linijka od dołu).

Komendy INKEY nie należy mylić z operatorem warunkowym INKEY, który jest częścią komendy IF.

Procedury

Istnieje możliwość zdefiniowania w pliku wsadowym czegoś w rodzaju procedury. Funkcjonalnie taka procedura odpowiada podprogramowi w Atari BASIC-u lub procedurze w Turbo-BASIC-u XL. Definicję procedury otwiera słowo kluczowe PROC, a zamyka ją słowo kluczowe RETURN, w następujący sposób:

```
PROC nazwa
```

...
RETURN

Procedurę wywołuje się przez podanie:

GOSUB nazwa

Przez GOSUB można także wywoływać etykiety zdefiniowane jak dla GOTO, pod warunkiem, że tak oznaczona sekwencja komend kończy się słowem RETURN.

Wywołania procedur mogą być zagnieżdżane, największa możliwa liczba zagnieżdżeń wynosi 20, ale może zostać zmniejszona przez inne czynniki, dlatego nie zaleca się przekraczania ośmiu poziomów zagnieżdżeń.

Inne polecenia

EXIT powoduje natychmiastowe zakończenie wykonywania pliku wsadowego. Dodatkowo można podać kod statusu wykonania, który system przyjmie jako numer błędu. Na przykład EXIT 170 spowoduje zakończenie pliku wsadowego z błędem 170 (File not found).

SETERRNO zapisuje podaną wartość w zmiennej systemowej ERRNO. Na przykład:

```
SETERRNO 170
```

ustawi 170 jako kod błędu, jaki ostatnio wystąpił w systemie. Najbardziej oczywistym zastosowaniem tej komendy jest skasowanie zmiennej ERRNO przed wykonaniem polecenia, którego wynik zamierzamy skontrolować przez IF ERROR.

Uwagi końcowe

Podczas wykonywania pliku wsadowego interpreter poleceń DOS-u (COMMAND.COM) jest cyklicznie ładowany do pamięci i z niej usuwany (nie jest programem rezydentnym), co nie pozostaje bez wpływu na szybkość interpretacji pliku BAT. Proces ten można przyspieszyć przez zatrzymanie interpretera w pamięci. Osiąga się to podając komendę LOAD COMMAND.COM na początku pliku poleceń. Usunięcie interpretera (np. przed zakończeniem wykonywania pliku wsadowego) osiąga się podając samo LOAD.

Rozdział 6: Programowanie ze SpartaDOS X

Ustawienie pozycji w pliku – POINT

Wewnętrzna procedura SpartaDOS X realizująca tę funkcję została napisana od nowa, dzięki czemu zmiany pozycji odczytu i zapisu w bardzo długich (ponad 500k) plikach powinny się wykonywać nawet do czterech razy szybciej niż do tej pory.

Odczyt pliku binarnego (LOAD)

Składnia: XIO 40,#IOCB,4,X,"D:[path]fname.ext"

Gdy X jest 0, plik zostanie załadowany do pamięci i uruchomiony. Gdy X ma wartość 128, plik zostanie załadowany do pamięci bez uruchamiania.

Odczyt danych dyskietki (CHKDSK)

Działanie tej funkcji uległo drobnej zmianie w związku z poszerzeniem repertuaru obsługiwanych gęstości dysków. Poprzednie wersje SpartaDOS obsługują dyski z sektorami 128- i 256-bajtowymi – SpartaDOS X v.4.40 dodaje do tego obsługę dysków z sektorami o wielkości 512 bajtów i (teoretycznie) więcej.

Funkcja CHKDSK zwraca w buforze 17 bajtów wyniku, ich znaczenie opisane jest w instrukcji do SpartaDOS X 4.20. Bajt kodujący wielkość sektora, znajdujący się pod adresem *bufor+1*, ma wartość 128 dla gęstości 128 BPS (czyli pojedynczej i średniej) oraz 0 dla 256 BPS (podwójnej). W intencji twórców SpartaDOS-u jest to oczywiście wartość młodszego bajtu liczby określającej wielkość sektora (128 = \$0080, 256 = \$0100), jednak w ten sposób przekazanie informacji o tym, że sektor ma np. 512 bajtów jest niemożliwe.

Dlatego też w SpartaDOS X v. 4.40 interpretacja tego bajtu, zarówno wewnątrz samego DOS-u, jak i w programach użytkowych dostarczanych wraz z nim, została zmieniona w taki sposób, żeby program mógł rozpoznać inne rozmiary sektora niż tylko 128 i 256, a jednocześnie, żeby została zachowana kompatybilność wstecz.

Wartość 128 w tym bajcie oznacza, jak poprzednio, 128 BPS. Natomiast każda inna to wartość starszego bajtu wielkości sektora logicznego *zmniejszona o 1*. Pozwala to na łatwe zakodowanie wielkości sektorów od 128 bajtów do 64 kilobajtów, według następującej zasady (* – wielkości nieobsługiwane):

Wielkość	Wartość hex	Kodowanie Uwagi
----------	-------------	-----------------

128 B	\$0080	\$80 (128)	Dla zgodności wstecz.
256 B	\$0100	\$00 (0)	Jak w formacie v.2.0.
512 B	\$0200	\$01 (1)	
*1024 B	\$0400	\$03 (3)	
*2048 B	\$0800	\$07 (7)	
*4096 B	\$1000	\$0F (15)	
*8192 B	\$2000	\$1F (31)	
*16384 B	\$4000	\$3F (63)	
*32768 B	\$8000	\$7F (127)	
*65536 B	\$0000	\$FF (255)	

Podprogram w asemblerze przeliczający wartość z kolumny „Kodowanie” powyżej na dwubajtową wielkość sektora właściwą dla DCB może wyglądać np. tak:

```

;wartosc kodu podajemy w akumulatorze (A)
;wynik odbieramy w A (mlodszy bajt)
;i X (starszy bajt)
;
getssize
    ldx #$00
    cmp #$80
    beq qu
    tax
    inx
    lda #$00
quit    rts

```

Tablica danych SpartaDOS (COMTAB)

DECOUT2 COMTAB-21

Zawiera uzupełniony spacjami, przesunięty w prawo wynik procedury *misc_conv32*, czyli ciąg znaków ASCII reprezentujący czterobajtową liczbę z DIVEND (zob. Wartości tablicy na stronie 7). 10 bajtów (w tym osiem bajtów DECOUT).

DIVEND COMTAB-6

Zawarta tu czterobajtowa liczba jest zamieniana przez procedurę *misc_conv32* na tekst umieszczany w DECOUT2.

ODATER COMTAB+19

OTIMER COMTAB+22

TDOVER COMTAB+25

W działaniu podobne do rejestrów (kolejno) DATE, TIME i DATESET, z tym jednak wyjątkiem, że TDOVER w przeciwieństwie do DATESET nie jest automatycznie kasowany po jednokrotnym użyciu.

ODATER, OTIMER i TDOVER to są stare rejestry kontroli czasu i daty plików wykorzystywane w SpartaDOS 3.x. W SpartaDOS X 4.20 „wyłączono” je i zastąpiono rejestrami DATE/TIME/DATESET. Jest to

przyczyna, dla której programy kopiujące ze SpartaDOS Construction Set uruchomione ze SpartaDOS X 4.2x nie są w stanie odtworzyć w kopii czasu i daty kopiowanego pliku. W SpartaDOS X 4.40 funkcję tych rejestrów przywrócono, DATESET ma jednak priorytet, i gdy jest ustawiony, stan TDOVER zostanie zignorowany przez DOS.

W SpartaDOS X 4.40 TDOVER jest zerowany po zakończeniu programu, ale w SpartaDOS 3.x nie, dlatego do dobrego stylu programowania należy wyzerowanie tego rejestru przed wyjściem do DOS-u (oczywiście tylko wtedy, kiedy program go do czegoś używał).

Dekodowanie identyfikatora dysku

Jak wspomniano w poprzednich sekcjach suplementu, jedną z nowych możliwości, jakie oferuje SpartaDOS X 4.40, jest zwiększona liczba możliwych do użycia identyfikatorów dysków. Mówiąc prościej, gdy inne DOS-y obsługują osiem lub dziewięć dysków (ponumerowanych od D1: do D9:), SpartaDOS X 4.40 obsługuje ich piętnaście. „Nadprogramowe” dyski (powyżej D9:) oznaczone są kodami literowymi od DJ: do DO:, zgodnie z konwencją przyjętą już w poprzednich wersjach SpartaDOS, że stacja nr 1 może być oznaczona przez D1: albo przez DA:, stacja nr 2 to D2: lub DB: i tak dalej.

W przypadku gdy program posługuje się przy wywołaniach funkcji DOS-u identyfikatorem dysku lub specyfikacją pliku sformatowaną przez SpartaDOS (np. odczytaną z linii poleceń), nie będzie raczej żadnych problemów z dostępem do dysków spoza tradycyjnego zakresu. Kłopotu nie powinno też być, kiedy program zechce z otrzymanej specyfikacji coś wyliczyć (np. wartość zmiennej DUNIT w DCB), mimo że w innych DOS-ach po literze „D”, a przed dwukropkiem znajduje się cyfra dziesiętna ASCII (czyli wartość z zakresu od \$31 do \$39), w SpartaDOS X natomiast może to być równie dobrze litera – czyli wartość ASCII z zakresu od \$41 do \$4F. W celu otrzymania „binarnego” numeru dysku trzeba – rozkazem AND #\$0F – „obciąć” starszy półbajt, a prostota tej operacji powoduje, że prawie każdy programista tak robi.

Konwersja w drugą stronę może wszelako nastreczyć kłopotów, gdyż w celu uzyskania poprawnej wartości ASCII z „binarnego” numeru dysku trzeba dodać \$30 dla liczb z zakresu od 1 do 9, a dla większych - \$40. Ponieważ do tej pory rozróżnianie między „od 1 do 9” a „więcej” nie było potrzebne, programy dokonujące w jakichś, sobie tylko znanych celach takiego działania prawdopodobnie nie będą poprawnie działały z dyskami od 10 do 15. Na szczęście takie przeliczenie rzadko bywa potrzebne – w razie czego jednak poniżej załączamy przykład:

```
dsk2asc
    lda dunit
    ora #$30
    cmp #'9+1
```

```

    bcc ok
    adc #$0f
ok   rts

```

Wynik – znak ASCII symbolizujący dany dysk – zwracany jest w akumulatorze. Będzie to działać z każdym DOS-em. Natomiast jeśli program jest przeznaczony tylko dla SpartaDOS X, procedurkę powyższą można znacznie uprościć:

```

dskzasc
    lda dunit
    ora #$40
    rts

```

Symbole

Symbol jest ośmioznakową nazwą obiektu znajdującego się gdzieś w pamięci komputera. Obiektem tym może być struktura danych lub procedura. Znając nazwę symbolu możemy w programie przełożyć ją na konkretny adres. Zwykle programy binarne muszą robić to „na piechotę”, to znaczy wywołując odpowiednią procedurę tłumaczącą nazwę symbolu na konkret (patrz „Wartości tablicy na stronie 7”). Nie jest to ani najwygodniejszy, ani jedyny sposób; niektóre asemblery umieją generować przeznaczone dla SpartaDOS X binaria o specjalnej strukturze – w takim przypadku „tłumaczenie” symboli na adresy robione jest automatycznie przez loader binarny SpartaDOS.

Obecność symbolu świadczy o tym, że odpowiednia procedura jest załadowana do pamięci, sam zaś symbol informuje, gdzie się ona znajduje. Adresy wskazywane przez symbole zmieniają się w zależności od wersji SpartaDOS oraz od kolejności ładowania nakładek. Część symboli wskazuje procedury zdefiniowane w module ROM, a nawet same symbole bywają tam zapisane, nie należy jednak nawet wtedy liczyć na ich stałość – każdy symbol bowiem może zostać w każdej chwili zastąpiony przez symbol o takiej samej nazwie, jednak wskazujący inne miejsce w pamięci. Tak dzieje się na przykład wtedy, kiedy nakładka wczytana z dysku zastępuje własnym kodem którąś z procedur systemowych.

Brak symbolu oznacza na ogół, że nie wczytano odpowiedniego sterownika.

Wartości pod OS ROM

Wektory pod OS ROM są umieszczane celem zapewnienia zgodności z oprogramowaniem pisanim pod SpartaDOS 3.2 i wcześniejsze wersje. Pod SpartaDOS X nie ma potrzeby z nich korzystać, gdyż te same procedury wskazywane są przez odpowiednie symbole, przeto nie ma potrzeby zaglądnienia pod ROM ani kłopotania się, czy umieszczonych

tam przez system wektorów coś nie nadpisało.

Oto lista symboli zastępujących stare wektory:

Wektor	Nazwa	Symbol	Definiowany przez
-----	-----	-----	-----
\$FFC0	VGETTD	I_GETTD	sterowniki zegara (CLOCK.SYS)
\$FFC3	VSETTD	I_SETTD	jak wyżej
\$FFC6	VDON	I_TDON	TD.COM
\$FFC9	VFMTTD	I_FMTTD	TD.COM
\$FFCC†	VINITZ	_INITZ	kernel
\$FFCF†	VINITZ2	-	-
\$FFD2	VXCOMLI	-	-
\$FFD5†	VCOMND	-	-
\$FFD8†	VPRINT	PRINTF	kernel
\$FFDB†	VKEYON	I_KEYON	KEY.COM

Wektory pod ROM-em nie są używane przez sam SpartaDOS, mogą też zostać zniszczone przez uruchamiane programy (np. przez Turbo BASIC XL), co czyni korzystanie z nich niepewnym i kłopotliwym. W przyszłych wersjach SpartaDOS X mogą zniknąć zupełnie (w tabeli krzyżykiem oznaczono wektory nie działające już w SpartaDOS X 4.20).

Działanie symboli I_GETTD, I_SETTD, I_TDON i I_KEYON jest takie samo, jak działanie procedur VGETTD, VSETTD, VTDON i VKEYON opisane w podręczniku użytkownika SpartaDOS X 4.20; z tym tylko, że fakt braku obsługującej wywołanie nakładki poznajemy nie po stanie bitu C po powrocie z procedury, lecz po niemożności jej wywołania spowodowanej brakiem symbolu.

W procedurach I_GETTD i I_SETTD ustawiony bit C sygnalizuje chwilową blokadę (stan zajętości) sterownika zegara. W takim razie należy odpowiednią procedurę wywołać jeszcze raz. Dobrą praktyką jest zliczanie takich błędnych wywołań i przerywanie pętli, gdy licznik osiągnie jakąś dużą wartość (np. przekroczy 255 prób).

Procedura I_FMTTD jest wczytywana do pamięci razem z programem TD.COM, stanowi bowiem jego integralny składnik. Jako parametr wejściowy należy jej podać adres bufora o wielkości co najmniej 32 znaków: młodszy bajt w rejestrze Y, starszy w rejestrze X. Powrót procedury z ustawionym bitem C sygnalizuje błąd (permanentną blokadę zegara i niemożność odczytania czasu w związku z tym). Jeśli znacznik C jest skasowany, w buforze znajduje się informacja na temat bieżącego czasu i daty w postaci sformatowanego ciągu tekstowego, zakończonego znakiem Return (\$9B).

Wartości tablicy na stronie 7

1) Wektor *misc*

Zarówno oryginalna, angielska instrukcja do SpartaDOS X 4.20, jak i jej polskie tłumaczenie wykonane przez PZ Karen (str. 99) błędnie podają kod operacyjny funkcji *misc_convdc* jako 4, podczas gdy w rzeczywistości jest to 5. Niniejszy suplement daje dobrą okazję do skorygowania tego błędu.

Ważniejszą informacją może być, że znajdująca się „za” tym kodem funkcja konwertująca jest obecnie 32-bitowa, z najstarszym bajtem DIVEND znajdującym się pod adresem COMTAB-3 i generująca dziesięciocyfrowe wyniki w DECOUT2. Wszelako dla zachowania kompatybilności z istniejącym oprogramowaniem wejście *misc_convdc* przed dokonaniem obliczenia zeruje najstarszy bajt liczby przycinając ją do 24 bitów, a wynik do ośmiu cyfr. Do konwersji 32-bitowej służy funkcja *misc_conv32* o kodzie operacyjnym 11, która, co ważne, dostępna jest *tylko* w wersjach SpartaDOS X 4.40 i nowszych. Wywołanie jej na SpartaDOS 4.2x skończy się najprawdopodobniej zawieszeniem komputera.

2) Rejestr *device*

Kod urządzenia \$6x w rejestrze *device* przypisano do urządzenia NUL:

3) Procedura *block_io*

block_io \$0706

Procedura *block_io* służy do zapisu i odczytu sektorów. Użycie jej w tym celu zamiast *lsio* uwalnia programistę od ustawiania części zmiennych DCB oraz zmartwień o wielkość sektora, zmienna DBYT (\$0308-\$0309) jest bowiem ustawiana automatycznie na właściwą wartość w zależności od rozpoznanej – też automatycznie – gęstości oraz numeru sektora (w podwójnej gęstości, jak wiadomo, pierwsze trzy sektory mają po 128 bajtów, a reszta po 256).

Przed wywołaniem *block_io* w DCB należy ustawić: kod urządzenia (DDEVIC), numer stacji (DUNIT), adres bufora (DBUFA) i numer sektora (DAUX1/2), a numer wywoływanej funkcji w rejestrze Y. Po powrocie w akumulatorze znajduje się status operacji: ujemny kod błędu w przypadku niepowodzenia lub wartość dodatnia (0 albo 1) w przeciwnym razie.

Numer funkcji *block_io* są następujące:

- 0 – *bio_rdsec* – zwykły odczyt sektora (bez rozpoznania gęstości)
- 1 – *bio_wrsec* – zwykły zapis sektora (j.w.)
- 4 – *bio_rdsys* – odczyt sektora z rozpoznaniem i zapamiętaniem gęstości

5 – *bio_sbps* – zapamiętanie wielkości sektora zapisanej w DBYT

Pozostałe kody operacyjne (2 i 3) są zarezerwowane do wewnętrznego użytku procedury SPARTA.SYS.

Funkcje nr 0 i 4 działają tak samo, z tym że ta druga na początku odczytuje ze stacji informację o wielkości sektorów i zapisuje ją sobie w pamięci. Funkcje 0 i 1 korzystają potem z tej informacji przy „zwykłym” odczycie i zapisie sektorów – tak więc program, który realizuje bezpośredni dostęp do dysku tą drogą powinien funkcję nr 4 wywołać jako pierwszą, przy okazji np. wczytywania sektora nr 1, a do odczytu dalszych sektorów tego dysku posługiwać się funkcją 0.

Ewentualnie, jeśli zachodzi potrzeba ominięcia automatyki DOS-u, a wielkość sektora jest znana skądinąd, funkcję nr 4 można zastąpić przez wpisanie żądanej wielkości sektora do DBYT (\$0308-\$0309), a następnie wywołanie kolejno funkcji 5 i 0.

4) Procedura *fsymbol*

```
fsymbol  $07EB
ext_on   $07F1
ext_off  $07F4
```

Nowością w SpartaDOS X 4.40 jest procedura *fsymbol* (\$07EB) dokonująca tłumaczenia nazwy symbolicznej (zob. wyżej) na odpowiadającą jej informację o położeniu w pamięci (adres oraz kod rodzaju pamięci). Odszukanie symbolu za jej pomocą jest bardzo proste, w rejestrach AX (młodszy/starszy) należy podać adres nazwy symbolu. Nazwa ta musi mieć zawsze osiem znaków, jeśli jest krótsza, dopełniamy ją spacjami.

Jeśli po wywołaniu *fsymbol* ustawiony jest znacznik Z rejestru znaczników procesora, oznacza to, że symbol nie jest zdefiniowany. W przeciwnym wypadku rejestry AX zawierają adres wskazywany przez symbol, a rejestr Y – kod rodzaju pamięci, w której znajduje się poszukiwany obiekt. Kod ten należy przekazać (w akumulatorze) procedurze *ext_on*, dokonuje ona odpowiedniego przełączenia banków pamięci (o ile to potrzebne). W tym momencie można już użyć adresu zwróconego przez *fsymbol* – a po zakończeniu trzeba posprzątać wywołując procedurę *ext_off*, która przywraca stan pamięci sprzed wywołania *ext_on*.

5) Tablica *sio_index* (swap)

```
sio_index    $070F
```

Tablica *sio_index* zmieniana przez polecenie SWAP oraz używana

przez sterowniki SIO ma, mimo zwiększenia liczby napędów do 15, wciąż tylko 9 pozycji i nie może być poszerzona – dlatego napędy o numerach od 10 do 15 nie mogą być „swapowane”.

Rozdział 7: Informacje techniczne

Sektory odczytu wstępnego

W standardowych gęstościach Atari, to znaczy 128 i 256 BPS, trzy pierwsze sektory mają po 128 bajtów i są zajęte przez program dokonujący odczytu wstępnego. Program ten nie uległ zasadniczym zmianom w SpartaDOS X v. 4.40 i jest prawie identyczny, jak w poprzednich wersjach.

Nowością jest to, że gdy sektory danych na dysku są większe niż 256 bajtów, pierwsze trzy sektory mają taką samą wielkość, jak wszystkie inne. Co więcej, obszar odczytu wstępnego obejmuje tylko jeden sektor, ten o numerze 1. Pierwsze 42 bajty przeznaczone są na informacje o dysku, tak samo jak w zwykłym formacie SpartaDOS. Pozostałą część sektora wypełnia program bootujący przystosowany do obsługi sektorów 512-bajtowych i większych.

Może się wydawać, że taka organizacja dysku (i de facto zerwanie z ustalonym standardem) to niepotrzebna komplikacja. Faktycznie, w samym SpartaDOS X i przeznaczonych dla niego programach narzędziowych musiały zajść głębokie zmiany w mechanizmie rozpoznawania gęstości dysku, bazującym dotąd na fakcie, że wielkość pierwszego sektora jest znana z góry, i że jest to 128 bajtów, a wielkość pozostałych sektorów ustala się na podstawie informacji zawartych w tymże sektorze numer 1. Drugim problemem jest to, że odczytu wstępnego (boota) z takiej „dyskiety” da się dokonać tylko z partycji twardego dysku: stary XL OS bowiem również nie potrafi prawidłowo ustawić wielkości sektora, co przy stacjach szeregowych oznacza błąd sumy kontrolnej przy próbie odczytu i niepowodzenie startu.

Niemniej w naszej ocenie nad niedogodnościami przeważają korzyści. Pierwszą z nich jest dostosowanie się do standardu pamięci masowych przyjętego na całym świecie, w którym wszystkie sektory na dysku są takiej samej wielkości, a najmniejszą możliwą jednostką alokacji danych jest sektor o pojemności 512 bajtów. Oznacza to łatwiejszy dostęp do produkowanych obecnie jednostek pamięci masowych. Drugą – przyrost ilości miejsca w obszarze odczytu wstępnego (z 384 do 512 bajtów), dzięki czemu umieszczenie tam programu zdolnego bootować z dysku o sektorach 512-bajtowych i większych było w ogóle możliwe. Nie do pogardzenia jest też fakt lepszego wykorzystania miejsca na nośniku – aczkolwiek w przypadku twardego dysku utratę nieco ponad 1 kilobajta można byłoby pewnie przeboleć.

Informacja o strukturze systemu plików

Informacja o filesystemie jest w zasadzie taka sama, jak w zwykłym formacie SpartaDOS, lista różnic znajduje się poniżej:

31 Rozmiar sektorów fizycznych na dyskietce (1 bajt): \$80 – 128 bajtów, \$00 – 256 bajtów, \$01 – 512 bajtów, pozostałe wartości są zarezerwowane. Ogólnie, wszystkie wartości różne od \$80 oznaczają starszy bajt rozmiaru sektora, odjąć 1.

32 Numer wersji filesystemu (1 bajt): SpartaDOS 2.x, 3.x i SpartaDOS X 4.2x ustawiają go na \$20, co oznacza wersję 2.0. SpartaDOS X 4.40 ma tu \$21 (wersja 2.1).

33-34 Od wersji 2.1 filesystemu: rozmiar sektorów fizycznych na dyskietce podany w bajtach, w standardowej konwencji młodszy/starszy (2 bajty).

35-36 Od wersji 2.1 filesystemu: liczba wpisów w sektorze mapy pliku.

37 Od wersji 2.1 filesystemu: liczba sektorów fizycznych przypadająca na jeden sektor logiczny (klaster). W chwili obecnej DOS akceptuje tu tylko \$01.

W SpartaDOS X 4.40 wartości te są tylko do odczytu, nie można ich zmieniać. Bajty 42-63 bootsektora są zarezerwowane dla przyszłych rozszerzeń i ich wartości nie powinny być zmieniane.

UWAGA: Bajty 33-37 mają inne znaczenie w systemie plików SpartaDOS 1.1, a w późniejszych wersjach, mimo że nie są używane, zachowują wartości domyślne dla 1.1. Z tego powodu przed skorzystaniem z zawartych tu informacji programista musi koniecznie zadbać o uprzednie sprawdzenie numeru wersji systemu plików!

Struktura katalogu

Struktura katalogu jest identyczna jak w poprzednich wersjach SpartaDOS. Jediną różnicą jest to, że pierwszy wpis (nagłówek) katalogu głównego ma wpisany czas i datę. Jest to czas i data zapisania świeżego katalogu na tym dysku.

Bezpośredni dostęp do dysku

Niektóre programy potrzebują dostępu do dysku z pominięciem DOS-u – np. kopiiery sektorowe – a jak nadmieniono powyżej, od wprowadzenia gęstości *DD 512* nie można już liczyć na to, że ustawienie ilości danych w DCB w ciemno na 128 bajtów i zażądanie odczytu sektora numer 1 przyniesie spodziewany efekt. Do ustalenia bieżącej konfiguracji dysku, z której da się „wywróżyć” wielkość sektora nr 1, trzeba posłużyć się komendą READ PERCOM.

Poniżej znajduje się przykład podprogramu, który zwraca w rejestrach AX (młodszy/starszy) wielkość sektora nr 1 dla stacji dysków o numerze podanym w akumulatorze:

```

ddevic      = $0300
dunit       = $0301
dcmnd       = $0302
dstats      = $0303
dbufa       = $0304
dtimlo      = $0306
dbyt        = $0308
daux1       = $030a
daux2       = $030b
jsioint     = $e459
buffer      = $0400      ;bufor magnetofonu

getbootsize
    sta dunit
    lda #$31
    sta ddevic
    lda #'N      ;komenda READ PERCOM
    sta dcmnd
    lda #$40
    sta dstats
    lda #<buffer
    sta dbufa
    lda #>buffer
    sta dbufa+1
    lda #$07
    sta dtimlo
;ilosc danych: 12 bajtow
    lda #$0c
    sta dbyt
    lda #$00
    sta dbyt+1
;to trzeba wyzerowac ze wzgledu na
;niektore systemy turbo dla stacji
;dyskow (np. Top Drive, TOM5 Turbo)
    sta daux1
    sta daux2
    jsr jsioint
    bpl success
;error 139 oznacza, ze stacja nie zna
;komendy READ PERCOM czyli
;moze dzialac tylko w 128 BPS (jest to
;Atari 810 lub 1050)
    cpy #139
    beq a810
;inny blad oznacza problemy z
;komunikacja
    cpy #$00
    rts
;nieprzerobiona stacja Atari 810 lub
;1050, 128 bajtow
a810    lda #$80
        ldx #$00
        ldy #$01
        rts
success
;mlodszy bajt rozmiaru sektora danych
    lda buffer+7
;starszy bajt rozmiaru sektora danych
;(odwrotna kolejnosc)
    ldx buffer+6
;jesli sektor < 512 bajtow, zwracamy 128
    cpx #$02
    bcc a810
;w przeciwnym razie tyle, ile odczytano
    ldy #$01
    rts

```

W praktyce rozsądniej jest odczytać z PERCOM i zapamiętać w programie wielkość „właściwych” sektorów danych, a ustalanie wielkości dla konkretnych sektorów, o ile jest potrzebne, zautomatyzować wewnątrz jednego podprogramu pośredniczącego w odczycie sektorów pomiędzy systemem operacyjnym a resztą programu.

Rozszerzenie bloku PERCOM

SpartaDOS X 4.40 uwzględnia rozszerzenie standardu PERCOM. W piątym bajcie bloku PERCOM (PERCOM+5), bit 3 ma następujące znaczenie: jeśli jest ustawiony (1), to dysk nie ma stron ani głowic, a bajt 4 bloku PERCOM (PERCOM+4) zawiera najstarszy bajt 24-bitowej liczby sektorów na ścieżkę. W przeciwnym wypadku, gdy ten bit jest skasowany, wartość bajtu 4 powinna zostać zignorowana dla partycji twardego dysku, tj. gdy liczba ścieżek, jaką podaje partycja, wynosi 1. Przyczyną jest to, że niektóre antyczne dyski SCSI do Atari zwracały w tym bajcie fizyczną liczbę głowic napędu.

Rozdział 8: Konfiguracja systemu.

Jedną z niepożądanych cech nowej wersji SpartaDOS X jest zwiększone zapotrzebowanie na pamięć. W związku z tym zaleca się użytkowanie DOS-u na komputerach wyposażonych w co najmniej 128k pamięci RAM oraz konfigurowanie go do pracy w trybie korzystania z rozszerzenia (USE BANKED w CONFIG.SYS).

Generalnie wskaźnik początku wolnej pamięci (MEMLO) nie powinien nigdy przekraczać wartości \$2000 w momencie uruchamiania większości programów – gdy jego wartość jest większa, prawdopodobieństwo wystąpienia błędu nr 179 (Memory conflict) wzrasta bardzo znacznie. Należy na to uważać przy instalowaniu nakładek, zwłaszcza w trybie USE OSRAM z buforami umieszczonymi w głównej pamięci.

Jeśli komputer nie ma rozszerzonej pamięci lub ma być ona wykorzystana do innych celów, najlepszym rozwiązaniem jest umieszczenie buforów DOS-u pod ROM-em systemu operacyjnego (tryb USE OSRAM / DEVICE SPARTA OSRAM w CONFIG.SYS). W takim układzie wskaźnik MEMLO pozostaje na rozsądnie niskim poziomie (w okolicach \$1100 po załadowaniu SPARTA.SYS i SIOOLD.SYS) i można zainstalować więcej nakładek.

Przy ocenie wartości MEMLO trzeba też wziąć pod uwagę fakt, że wskaźnik ten podnoszony jest również przez program X.COM, który jest niezbędny (w sposób jawny albo nie) do uruchomienia większości programów. Dlatego najlepiej po załadowaniu systemu wykonać polecenie LOAD X.COM i dopiero wtedy sprawdzić wartości wskaźników pamięci (poleceniem MEM).

Zestawy znaków

Gdy SpartaDOS X jest skonfigurowany do użycia pamięci pod ROM-em systemu operacyjnego jako miejsca przeznaczonego na bufor (USE OSRAM / DEVICE SPARTA OSRAM w CONFIG.SYS), może pojawić się problem z międzynarodowym zestawem znaków. Jest on wprowadzany do pamięci RAM, żeby przy przełączaniu banków pamięci nie pojawiały się na ekranie żadne zakłócenia, jednak ten sam obszar RAM-u jest też przeznaczony na bufor. Gdy jest ich za dużo – a konkretnie, więcej niż 6 – font zostanie zamazany i podczas operacji I/O na ekranie zobaczymy nieszkodliwe wprowadzanie, ale niezbyt estetyczne „krzaczkę”.

Wyjściem z sytuacji jest taka konfiguracja DOS-u, żeby liczba buforów znajdujących się pod ROM-em systemu operacyjnego nie przekraczała sześciu (domyślną wartością jest 4).

Konfiguracja domyślna

Standardowy plik CONFIG.SYS znajduje się obecnie na urządzeniu CAR: (CAR:CONFIG.SYS), skąd jest odczytywany, jeśli użytkownik nie zdefiniuje własnego na dysku. Dodatkowo zmienne środowiskowe CAR, BASIC i TEMP definiowane są przez sterownik RAMDISK.SYS po jego uruchomieniu, pod warunkiem wszakże, że użytkownik nie zdefiniował ich przedtem inaczej.

Zmienna COMSPEC nie jest ustawiana, jednak jej znaczenie i funkcja pozostają takie same, jak w poprzednich wersjach SpartaDOS X (wskazuje plik, który ma być załadowany zamiast CAR:COMMAND.COM).

Linia zaczynająca się od średnika (;) traktowana jest jako komentarz i ignorowana.

Słowo kluczowe MERGE

Od wersji SpartaDOS X 4.40 interpreter pliku CONFIG.SYS rozpoznaje nowe słowo kluczowe: MERGE. Jego użycie nie jest obowiązkowe, ale *jeśli ta komenda występuje w pliku CONFIG.SYS, musi być ostatnim słowem kluczowym tego pliku*, bo jej wystąpienie przerywa jego interpretację i „dołącza” (ang. *merge*) sobie inny. Pozwala to na uformowanie łańcucha kolejno wczytywanych plików, które razem składają się na dane o konfiguracji systemu. Przykład:

```
USE BANKED  
MERGE DEFAULTS
```

DOS skonfiguruje pamięć, a potem załadowuje plik o nazwie DEFAULTS.CFG. Ten plik powinien zawierać komendy konfiguracyjne – i może też mieć kolejną komendę MERGE na końcu, jeśli trzeba dołączyć następny kawałek skądinąd.

Jest to przydatne przy zarządzaniu większą liczbą konfiguracji przez Config Selector (opisany poniżej). Pliki, które się różnią tylko na początku, mogą dołączyć sobie resztę komend ze wspólnego źródła, a to pozwala na zmniejszenie liczby plików do wyedytowania przy globalnych zmianach konfiguracji.

Ograniczeniem MERGE jest to, że dołączany plik musi się znajdować w tym samym katalogu, co plik dołączający. Nie można też użyć MERGE zamiast słowa kluczowego USE, musi ono wciąż występować jako pierwsze. Rozszerzenie nazwy pliku (*.CFG) jest opcjonalne.

Config Selector

SpartaDOS X v. 4.40 ma wbudowany tzw. *config selector*, pozwalający na zdefiniowanie przez użytkownika kilku (do dziewięciu)

plików konfiguracyjnych i dokonanie przy starcie systemu wyboru, którego z nich użyć zamiast standardowego CONFIG.SYS.

Podczas startu systemu SpartaDOS X przeszukuje katalog główny dysku startowego na okoliczność obecności katalogu o nazwie SPARTA.DOS. Gdy jest on na dysku, sprawdza się jego zawartość pod kątem obecności plików z rozszerzeniem CFG. W razie ich znalezienia na ekranie pokazuje się menu pozwalające klawiszami cyfr (od 1 do 9, w zależności od liczby plików CFG w podkatalogu) wybrać któryś z nich, albo użyć standardowego pliku CONFIG.SYS (na dysku, jeśli jest, lub na urządzeniu CAR:). To ostatnie uzyskuje się po naciśnięciu klawisza **Esc** lub odczekaniu 5 sekund.

Żeby ten mechanizm zadziałał, dysk musi być w formacie SpartaDOS.

Programy systemowe

W procedurach systemowych (driverach) SpartaDOS X zaszło również sporo zmian, dodano także nowe programy tego typu. Te nowości zostaną teraz omówione po kolei.

1) OBSŁUGA SYSTEMU PLIKÓW

Procedura SPARTA.SYS

Struktura dla pojedynczego pliku z określanych przez parametr *nfiles* uległa powiększeniu z 35 do 40 bajtów. Pozostałe parametry bez zmian (minimum 2, maximum 16, standardowo 5).

W związku z obsługą sektorów 512-bajtowych rozmiar buforów (parametr *nbufs*) został zwiększony do 512 bajtów. Ich maksymalna liczba jednocześnie uległa ograniczeniu w niektórych konfiguracjach: USE NONE dopuszcza maximum osiem buforów, tak samo jest w przypadku umieszczenia buforów pod ROM-em systemu operacyjnego. Przy pracy z buforami w pamięci rozszerzonej (USE BANKED) lub głównej parametry pozostają te same, co w poprzednich wersjach SpartaDOS X: buforów może być nie więcej niż 16 i nie mniej niż 3, a domyślnie są cztery.

Trzeba zwrócić uwagę, że 16 buforów po 512 bajtów zajmuje dwa razy tyle pamięci, co analogiczna liczba buforów 256-bajtowych. Zadeklarowawszy 16 buforów w dodatkowej pamięci mamy więc wszelkie szanse na wypełnienie banku systemowego do ostatka – a w takiej sytuacji nakładki będą ładowane do głównej pamięci podnosząc MEMLO. Rozsądnie jest więc nie deklarować więcej niż 12 buforów, o ile naprawdę nie ma takiej potrzeby.

Podręcznik użytkownika SpartaDOS X 4.20 w tej sekcji niezbyt adekwatnie opisuje rolę owych powyżej wspomnianych „buforów”. Nie są to, jak w innych DOS-ach, bufory sztywno przypisane do otwartych

plików, lecz rodzaj pamięci podręcznej (*buffer cache*), w której procedura SPARTA.SYS przechowuje ostatnio używane sektory, bez różnicy, czy są to sektory odczytu wstępnego, mapy bitowej, map plików, czy sektory danych. Im „buforów” jest więcej, tym rzadziej SPARTA.SYS musi odwoływać się do dysku. Tak więc zmniejszenie ich liczby raczej nie spowoduje błędów, spowolni jedynie działanie systemu plików.

2) ODCZYT I ZAPIS SEKTORÓW

Procedura SIO.SYS

W SpartaDOS X v. 4.40 procedura SIO.SYS została znacznie ulepszona w stosunku do poprzednich wersji. Przede wszystkim stacja pracująca w protokole Ultra Speed jest na początku odpytywana, z jaką prędkością chciałaby prowadzić wymianę danych (stare SIO.SYS ustawiało na sztywno prędkość 52 kbps). Po drugie, raz ustawivszy tryb US, SIO nie daje się z niego zbyt łatwo „wybić”, dzięki czemu stacje TOMS mogą, jak to mają w zwyczaju, sypać błędnymi odpowiedziami na komendy, a transmisja pozostaje szybka. W razie, gdyby wybrany automatycznie tryb transmisji okazał się niewłaściwy, można go ręcznie zmienić komendą SIOSET.

Dodatkowo w SIO.SYS wbudowano mechanizm „mapowania” dysków, obsługiwany poleceniem MAP. Pozwala on zmienić numer logiczny napędu, przekierować obsługę I/O do systemu operacyjnego w ROM-ie lub wyłączyć dla danego napędu obsługę równoległego I/O (co w sytuacji, gdy mamy podpiętą partycję twardego dysku pod tym samym numerem, co szeregową stację dysków, spowoduje „wyłączenie” partycji twardego dysku).

Gdyby ktoś tego wszystkiego nie potrzebował, chciał natomiast zaoszczędzić nieco pamięci, na urządzeniu CAR: znajduje się również stara wersja SIO: SIOOLD.SYS. Jediną modyfikacją, jaka została wprowadzona w stosunku do tego, co było w poprzednich wersjach SpartaDOS, jest przystosowanie tego elementu systemu do pracy z 512-bajtowymi sektorami.

Procedura SIO2.SYS

Procedura jest miniaturową wersją sterownika urządzeń dyskowych (zamiennik dla SIO.SYS). Nie zawiera własnych procedur komunikacyjnych, lecz używa podsystemu SIO zawartego w systemie operacyjnym komputera. Parametry transmisji zależą więc od możliwości systemu operacyjnego. Oryginalny Atari OS jest pod tym względem dość ubogi, podczas gdy niektóre alternatywne systemy pozwalają na przyspieszenie transmisji czy obsługę nietypowych urządzeń.

SIO2.SYS nie może być używane, jeśli DOS jest skonfigurowany do używania pamięci pod ROM-em systemu operacyjnego (USE OSRAM).

Procedura CA2001.SYS

Zadanie: procedura szybkiej transmisji dla stacji California Access 2001.

Składnia: DEVICE CA2001 d:

Typ: zewnętrzny – na urządzeniu CAR.

Procedura ta nie zajmuje pamięci, lecz tylko programuje wskazaną stację California Access 2001 do pracy w szybkiej transmisji. Stacja pozostaje zaprogramowana aż do wyłączenia jej zasilania.

Procedura RAMDISK.SYS

Standardowo ramdysk zakładany jest jako stacja nr 15 (O:). Dodatkowo, jeśli użytkownik nie zdefiniował tego inaczej w pliku CONFIG.SYS, procedura RAMDISK.SYS zakłada zmienne środowiskowe CAR, BASIC i TEMP ustawiając je na „swoj” numer napędu.

Procedura PBI.SYS

Zadanie: dodatkowa obsługa urządzeń PBI.

Składnia: DEVICE PBI

Typ: zewnętrzny – na urządzeniu CAR:

Sterownik PBI.SYS wprowadza usprawnienia w obsłudze urządzeń dyskowych podłączonych do szyny równoległej. W przypadku niektórych programów występuje konflikt związany z odczytem danych do obszaru pamięci RAM pokrywającego się z procedurami obsługi urządzenia PBI. Sterownik likwiduje ten problem. PBI.SYS przyspiesza również transmisję w systemach wyposażonych w jedno urządzenie PBI. Uwaga: sterownik powoduje, że polecenia MAP oraz SIOSET nie będą działać dla dysków podłączanych przez PBI.

3) OBSŁUGA ZEGARA

Procedura ARCCLOCK.SYS

Zadanie: procedura obsługi sprzętowego zegara ARC

Składnia: DEVICE ARCCLOCK

Typ: zewnętrzny – na urządzeniu CAR:

Procedura obsługi sprzętowego zegara ARC (Atari Realtime Clock).

Procedura Z.SYS

Zadanie: zainstalowanie zgodnego wstecz ze SpartaDOS 3.2 urządzenia Z:

Składnia: DEVICE Z [/IS]

Typ: zewnętrzny – na urządzeniu CAR:

Program instaluje w tablicy handlerów OS sterownik urządzenia „Z:” dający prosty dostęp do funkcji odczytu/zapisu czasu i daty z poziomu np. BASIC-a. Sterownik ten jest kompatybilny z analogicznym sterownikiem istniejącym w SpartaDOS 3.2, tak więc stare oprogramowanie korzystające z niego nie powinno mieć (już) problemów z funkcjami czasu i daty pod SpartaDOS X.

Z.SYS ma cztery funkcje wewnętrzne wybierane komendami XIO BASIC-a (lub odpowiednimi dyrektywami systemu operacyjnego):

- 1) XIO 33: odczyt czasu w formie binarnej (read time, unformatted)
- 2) XIO 35: odczyt daty w formie binarnej (read date, unformatted)
- 3) XIO 36: ustawienie czasu (set time)
- 4) XIO 37: ustawienie daty (set date)

Procedura odczytu czasu jest następująca:

```
10 OPEN #1,4,0,"Z:":REM otworz do odczytu
15 REM ustaw na odczyt czasu
20 XIO 33,#1,4,0,"Z:"
25 REM pobierz stan zegara
30 GET #1,G:GET #1,M:GET #1,S
35 CLOSE #1
```

Ustawienie zegara:

```
10 OPEN #1,8,0,"Z:":REM otworz do zapisu
15 REM ustaw na zapis czasu
20 XIO 36,#1,8,0,"Z:"
25 REM przestaw zegar
30 PUT #1,G:PUT #1,M:PUT #1,S
35 CLOSE #1
```

Odczyt i ustawienie daty przeprowadza się analogicznie, jedynie kody operacyjne XIO trzeba zmienić na odpowiednio 35 i 37.

Próba odczytu lub zapisu ponad trzech bajtów będzie powodowała błąd nr 136 (EOF). Wyzerowanie znacznika odczytu i zapisu uzyskuje się przez zamknięcie i ponowne otwarcie kanału, albo przez ponowne wywołanie odpowiedniej funkcji XIO.

Funkcje ustawiania zegara są standardowo wyłączone, próby zapisu do „Z:” będą zwracać błąd 139 (NAK). Podanie w CONFIG.SYS

przełącznika /I (jak *ignore*) do Z.SYS sprawi, że te wywołania zamiast błędu będą zwracać status \$01 (sukces), ale nic poza tym nie będzie robione. Dopiero przełącznik /S (jak *set*) otwiera możliwość przestawienia zegara tą drogą.

Sterownik „Z:” obsługuje tylko jeden kanał we/wy – próba otwarcia drugiego spowoduje błąd nr 161 (Too many channels open).

Załadowanie nakładki TD.COM rozszerza sterownik „Z:” o dalsze funkcje:

- 5) XIO 38: włączenie wyświetlania czasu i daty (TD ON)
- 6) XIO 39: wyłączenie wyświetlania czasu i daty (TD OFF)
- 7) XIO 34: odczyt daty w postaci sformatowanego tekstu (read date, formatted)
- 8) XIO 32: odczyt czasu w postaci sformatowanego tekstu (read time, formatted)

Funkcje te będą działać tylko wtedy, gdy TD.COM jest załadowany do pamięci – w przeciwnym wypadku próby użycia będą zwracać status 139 (NAK). Przykład użycia:

```
10 DIM TIME$(13)
15 REM otwarcie do odczytu
20 OPEN #1,4,0,"Z:"
25 REM odczyt czasu, tekst sformatowany
30 XIO 32,#1,4,0,"Z:"
35 INPUT #1;TIME$
40 PRINT TIME$
45 CLOSE #1
```

Do zainstalowania Z.SYS wymagane jest uprzednie zainstalowanie procedury obsługi któregoś z zegarów, to jest CLOCK.SYS, ARCCLOCK.SYS lub JIFFY.SYS.

4) OBSŁUGA EKRAŃU

Procedura XEP80.SYS

Zadanie: instaluje sterownik urządzenia XEP80.

Składnia: DEVICE XEP80 [1|2] [/P|/N]

Typ: zewnętrzny – na urządzeniu CAR:

Sterownik może teraz pracować z urządzeniem podłączonym do dowolnego portu joysticka (określonego za pomocą pierwszego parametru wywołania, domyślnie 2). W poprzednich wersjach SpartaDOS X program XEP80.SYS zawierał błąd uniemożliwiający instalację na komputerach w wersji PAL. Obecna wersja poprawnie rozpoznaje takie komputery. Tryb działania można także wymusić za

pomocą opcji „/P” – PAL lub „/N” – NTSC.

Procedura QUICKED.SYS

Zadanie: przyspieszenie pracy edytora ekranowego.

Składnia: DEVICE QUICKED

Typ: zewnętrzny – na urządzeniu CAR:

QUICKED.SYS jest programowym akceleratorem operacji ekranowych, zastępuje on część procedur urządzeń CON: (DOS) oraz E: (OS), dzięki czemu zapis na ekran w trybie GRAPHICS 0 jest trzy do czterech razy szybszy niż normalnie.

Procedura CON64.SYS

Zadanie: 64-kolumnowy edytor ekranowy.

Składnia: DEVICE CON64

Typ: zewnętrzny – na urządzeniu CAR:

CON64.SYS to eksperymentalny program mający za zadanie emulować tryb tekstowy 64x24 przy użyciu trybu wyświetlania GRAPHICS 8 (320x192 piksele, tzw. hires). Po załadowaniu sterownika do pamięci, ekran pozostaje w zwykłym trybie 40-kolumnowym. Z poziomu interpretera poleceń DOS-u można wtedy wpisać komendę „CON64 ON” w celu przełączenia ekranu w tryb 64x24. Analogicznie, po „CON64 OFF” system przechodzi z powrotem w zwykły tryb tekstowy.

Tryb 64x24 nie jest zbyt przydatny do pracy z interpreterem poleceń. Po pierwsze, ekran zajmuje prawie 8k głównej pamięci, z wartością MEMTOP ustawioną na \$8035. Nie wszystkie programy dobrze to znoszą. Po drugie, nawet nie wszystkie programy narzędziowe SpartaDOS X są dostosowane do współpracy z takim trybem wyświetlania, np. MENU.COM i ED.COM będą mieć widoczne kłopoty. Część z tych problemów mamy nadzieję rozwiązać w przyszłych wersjach DOS-u.

Niemniej CON64.SYS może być z powodzeniem używany zarówno przy pracy w BASIC-u, jak i we własnych programach. Po zainstalowaniu sterownika urządzenie E: zyskuje dwie nowe funkcje:

- 1) XIO 64: włącza i wyłącza tryb 64-kolumnowy
- 2) XIO 65: pozwala stwierdzić, czy CON64.SYS jest załadowany do pamięci.

Gdy sterownik został poprawnie załadowany do pamięci, to niezależnie od tego, czy tryb 64-kolumnowy jest włączony, czy nie,

komenda XIO 65,#chn,12,0,"E:" powinna wykonać się bezbłędnie – albo zwrócić błąd 146 (Funkcja niezaimplementowana) w przeciwnym wypadku. „chn” to numer kanału IOCB otwartego dla konsoli, zwykle jest to 0 (w Atari BASIC zamiast 0 trzeba podać 16). Pozostałe wartości powinny być takie, jak w komendzie OPEN użytej do otwarcia kanału. GRAPHICS 0 to odpowiednik OPEN #0,12,0,"E:". Wywołanie XIO 65 zwraca informację na temat bieżącego stanu sterownika w ICAX5 (\$034E+IOCB*16): 128 jeśli tryb 64-kolumnowy jest włączony, 0 w przeciwnym wypadku.

Pomyślne wykonanie się XIO 65 sygnalizuje możliwość przełączenia w tryb 64-kolumn. Uzyskuje się to komendą XIO 64,#chn,12,0+m,"E:", gdzie parametr „m” może mieć wartość 128 (włączenie) lub 0 (wyłączenie). Poprzednia wartość „m” jest zwracana w ICAX3 (\$034C+IOCB*16).

64-kolumnowy edytor ekranowy działa tak samo (tylko nieco wolniej) jak zwykły edytor 40-kolumnowy. Największą różnicą jest wielkość linii logicznej: w trybie 64 kolumn składa się ona co prawda, tak samo jak w trybie 40-kolumnowym, z trzech linii fizycznych, ale ponieważ pojedyncza linia fizyczna jest dłuższa, linia logiczna ma wielkość do 192 bajtów (zamiast 120 poprzednio).

CON64.SYS przejmuje też pewne funkcje urządzenia S:. Gdy ekran jest pod jego kontrolą, w żadnym trybie graficznym nie będzie tradycyjnego okna tekstowego – sterownik 64-kolumnowy nie jest w stanie go utworzyć. W zamian tekst i grafika mogą być w miarę dowolnie mieszane na całym ekranie. Działanie trybów GRAPHICS 0, 8 i 24, gdy tryb 64-kolumnowy jest aktywny, jest następujące:

1) GRAPHICS 0: to jest „tekstowy” tryb 64x24. W tym trybie komenda BASIC-a POSITION będzie działać *tylko na kursor tekstowy*. Przepuszczalnie będzie można użyć PLOT i DRAWTO, ale nie jest to zalecane, bo oba urządzenia, S: i E:, będą działać w oparciu o te same współrzędne, a zatem jest prawdopodobne, że próba narysowania czegokolwiek za pomocą tego pierwszego spowoduje raportowanie błędu 141 (Kursor poza ekranem) przez to drugie.

2) GRAPHICS 24: to jest czysto graficzny tryb 320x192. W tym trybie komenda POSITION będzie działać *tylko na kursor graficzny*. Wyprowadzenie tekstu przez urządzenie E: przypuszczalnie się uda, ale nie zalecamy tego z podobnych powodów, jak te wyłuszczone powyżej.

3) GRAPHICS 8: to jest graficzny tryb 320x192 z „oknem tekstowym”. W tym trybie POSITION będzie działać *tylko na kursor graficzny*. Pozycja kursora tekstowego może być zmieniana przez ingerencję w zmienne systemowe TXTCOL (\$0291) i TXTROW (\$0290), odpowiednio dla współrzędnych *x* i *y*. W tym trybie można zupełnie swobodnie tak rysować jak i pisać, bo system przechowuje oddzielne zestawy współrzędnych dla ekranu graficznego (S:) i konsoli (E:).

Trzeba zwrócić uwagę, że wspomniane „okno tekstowe” nie jest ograniczone do trzech dolnych wierszy ekranu, lecz pokrywa cały obszar ekranu graficznego. Ma to ten efekt uboczny, że np. wyczyszczenie ekranu tekstowego (przez CLS) kasuje też grafikę, i odwrotnie.

CON64.SYS wymaga komputera XL/XE wyposażonego w rozszerzenie pamięci zgodne ze 130XE.

Procedura CON80.SYS

Zadanie: 80-kolumnowy edytor ekranowy.

Składnia: DEVICE CON80

Typ: zewnętrzny – na urządzeniu CAR:

CON80.SYS to eksperymentalny program mający za zadanie emulować tryb tekstowy 80x24 przy użyciu trybu wyświetlania GRAPHICS 8 (320x192 piksele, tzw. hires). W działaniu program jest bardzo podobny do opisanego powyżej CON64.SYS, tutaj zostaną więc tylko wyszczególnione różnice między nimi.

Po załadowaniu sterownika komendy do włączenia i wyłączenia trybu 80-kolumnowego to odpowiednio „CON80 ON” i „CON80 OFF”.
Funkcje dodawane do urządzenia E: to:

1) XIO 81: odpowiednik XIO 65 w CON64.SYS

2) XIO 80: odpowiednik XIO 64 w CON64.SYS

Wielkość linii logicznej to 240 bajtów. Niestety, interpretery BASIC-a mają problemy z wierszami tej wielkości, Turbo BASIC XL potrafi się nawet zawiesić. Dlatego bezpieczniej jest nie przekraczać 192 znaków na linię programu.

Emulowanie trybu 80-kolumnowego w grafice 320x192 jest dużo mniej skomplikowane i wymaga mniej obliczeń niż emulacja trybu 64-kolumnowego. To jest powód, dla którego CON80.SYS jest krótszy, a także szybszy, od CON64.SYS.

CON80.SYS wymaga komputera XL/XE wyposażonego w rozszerzenie pamięci zgodne ze 130XE.

5) OBSŁUGA KLAWIATURY

Procedura CAD.SYS

Zadanie: instaluje program „miękkiego” resetu.

Składnia: DEVICE CAD keycode repeat ON|OFF

Typ: zewnętrzny – na urządzeniu CAR:

Niektóre programy, np. Disk Communicator lub The First XLEnt

Wordprocessor nie oferują użytkownikowi możliwości powrotu do DOS-u. Nakładka CAD.SYS rozwiązuje ten problem: po jej zainstalowaniu, w przypadku konieczności wyjścia z programu, którego autor zapomniał o takiej opcji, wystarczy nacisnąć zdefiniowaną kombinację klawiszy, która „ubije” program i powróci do DOS-u.

Parametr *keycode* to jest kod klawiaturowy kombinacji klawiszy, która ma uaktywniać procedurę „miękkiego resetu”. Jest on identyczny z kodem skaningowym generowanym przez układ Pokey w rejestrze KBCODE. Zalecane wartości to \$E7 (Ctrl/Shift/Inverse) lub \$CC (Ctrl/Shift/Return).

Parametr *repeat* określa, ile razy trzeba kolejno nacisnąć wybraną kombinację klawiszy, żeby nakładka zadziałała. Zero oznacza 256 razy.

Ostatni parametr określa, czy po wciśnięciu przez użytkownika klawisza RESET należy ustawić klawiaturę w tryb liter dużych (ON) czy małych (OFF).

6) URUCHAMIANIE PROGRAMÓW

Procedura RUNEXT.SYS

Zadanie: powiązanie typów plików z programami aplikacyjnymi (obsługa tzw. skojarzeń plików).

Składnia: DEVICE RUNEXT [d:][path][filename.ext]

Typ: zewnętrzny – na urządzeniu CAR:

RUNEXT.SYS jest rozszerzeniem pozwalającym ustalać skojarzenia między plikami danych a programami, które mają je otwierać. Na przykład skojarzenie plików *.ARC z programem ARC.COM powoduje, że COMMAND.COM automatycznie uruchamia program ARC.COM, kiedy użytkownik wpisze jako komendę nazwę archiwum (wraz z rozszerzeniem) i naciśnie klawisz Return.

Opcjonalnym parametrem jest nazwa pliku konfiguracyjnego zawierająca skojarzenia. Gdy jej nie podano, zostanie załadowany plik CAR:RUNEXT.CFG.

Plik konfiguracyjny składa się z linii definiujących po jednym skojarzeniu oraz z linii komentarza (te ostatnie oznaczane są średnikiem lub gwiazdką na początku). Format linii definiującej skojarzenie jest następujący:

EXT,PROGRAM [,PARAMETERS]

gdzie:

EXT – 3-literowe rozszerzenie, do którego tworzymy skojarzenie

PROGRAM – nazwa programu, który kojarzymy z rozszerzeniem, poprzedzona ewentualną ścieżką

PARAMETERS – opcjonalne parametry przekazywane do programu; jeżeli je pominiemy, wówczas przekazywana jest tylko nazwa pliku z

rozszerzeniem; jeżeli w parametrach użyjemy znaku procenta (%), wówczas wystąpi ona we wskazanym miejscu.

Przykład:

ARC,CAR:ARC.COM,L %

W ten sposób określamy skojarzenie dla plików z rozszerzeniem „ARC”. Pliki takie otwierane będą za pomocą programu CAR:ARC.COM uruchamianego z parametrem L, po którym występuje nazwa pliku. W efekcie, wprowadzenie w linii poleceń nazwy archiwum, np.

D1:ARCHIVE.ARC

spowoduje wyświetlenie jego zawartości.

Przy zainstalowanym RUNEXT.SYS można tymczasowo zapobiec jego działaniu poprzedzając wprowadzane polecenie znakiem „+”.

Procedura COMEXE.SYS

Zadanie: automatyczne zarządzanie modułem SpartaDOS X przy uruchamianiu programów.

Składnia: DEVICE COMEXE

Typ: zewnętrzny – na urządzeniu CAR:

COMEXE.SYS jest to rozszerzenie systemu wprowadzające rozróżnienie pomiędzy plikami *.COM a plikami *.EXE. Gdy jest zainstalowane, pliki typu *.COM są traktowane jako polecenia zewnętrzne (*command*) i uruchamiane jak dotychczas. Natomiast co do plików z rozszerzeniem *.EXE, to przed ich uruchomieniem system automatycznie odłącza moduł biblioteki – innymi słowy, rozszerzenie *.EXE oznacza, że program trzeba uruchamiać „przez X”, i system to sam wykonuje. Przy uruchamianiu plików nie trzeba podawać rozszerzenia (COM lub EXE).

Przy zainstalowanym COMEXE.SYS można tymczasowo zapobiec jego działaniu poprzedzając wprowadzane polecenie znakiem „+”.

7) INNE

Program INIDOS.SYS

Zadanie: uruchamianie SpartaDOS X.

Składnia: brak.

Typ: program systemowy.

Wykonanie polecenia **COLD /N** w interpreterze poleceń, lub zimnego startu systemu podczas pracy z BASIC-em oraz większością programów

powoduje zupełną dezaktywację modułu SpartaDOS X. Ponowne jego uaktywnienie nie jest zwykle możliwe bez wyłączenia i ponownego włączenia zasilania komputera – a to powoduje na przykład utratę zawartości ramdysku.

Problem ten rozwiązuje program INIDOS.SYS. Należy go skopiować na dysk, z którego będzie uruchamiany komputer, a następnie wydać komendę **BOOT INIDOS.SYS**. Gdy zachodzi potrzeba uruchomienia SpartaDOS X po tym, jak moduł został unieczynniony komendą COLD / N, wystarczy – bez wyłączania komputera – spowodować zimny start systemu i zbootować komputer z tego dysku.

Rozdział 9 – Użycie innych programów

BASIC XE

W trybie korzystania z pamięci położonej pod ROM-em systemu operacyjnego (USE OSRAM), z braku wystarczającej ilości miejsca tamże, SpartaDOS X wykorzystuje teraz do własnych celów pamięć RAM znajdującą się w obszarze \$D800-\$DFFF. W to samo miejsce ładują się rozszerzenia BASIC-a XE, a więc informacja zamieszczona w Podręczniku Użytkownika SpartaDOS X, że można używać BASIC-a XE z rozszerzeniami w trybie USE OSRAM, nie jest już aktualna – wymaganym trybem do tego jest USE BANKED, a komputer musi mieć ponad 128k RAM.

DiskRx

Dedykowany dla SpartaDOS edytor dyskowy DiskRx nie jest przystosowany do gęstości *DD 512* i nie może być wykorzystany do edycji dysków w niej sformatowanych. Ponieważ kod źródłowy jest niedostępny, nie było na to innej rady, jak napisanie podobnego programu od nowa: edytor *Eddy* jest dostępny w internecie pod adresem <http://drac030.krap.pl/pl-sparta-pliki.php>.

Drugim problemem z programem DiskRx jest to, że sprawdza on typ systemu plików dysku, z którego jest wczytywany, i odmawia współpracy, jeśli nie jest to dysk w formacie SpartaDOS. Filesystem zapisany w sektorach 512-bajtowych nie jest rozpoznawany przez tę procedurę, w związku z czym edytora DiskRx w wersji 1.9 nie można z takiego dysku nawet uruchomić. Istnieje poprawiona wersja *1.9a*, w której nie występuje ten kłopot. Wciąż jednak nie można jej używać do edycji dysków z sektorami 512-bajtowymi.

CleanUp

Program CleanUp nie jest przystosowany do gęstości *DD 512* i nie może być wykorzystany do sprawdzania poprawności struktury plików na nich zapisanej.

Dodatek A

Komunikaty błędów – w SpartaDOS X v. 4.40 zaszły tu następujące zmiany:

148 – komunikat zmieniono z „Unrecognized diskette format” na „Unknown filesystem”.

166 (Range error) – podczas operacji plikowych oznacza: przy odczycie próbę odczytu poza końcem pliku; przy zapisie: przekroczenie maksymalnej dozwolonej wielkości pliku (16 MB dla pliku lub 32 kB dla katalogu). Ogólnie: parametr operacji jest poza dopuszczalnym zakresem.

169 „Directory full” – nie można utworzyć pliku, bo katalog jest już całkowicie wypełniony. Katalog w SpartaDOS X mieści maksymalnie 1423 wpisy plików i podkatalogów. W poprzednich wersjach DOS-u przekroczenie tego limitu powodowało błąd 162 i komunikat „Disk full”, niezbyt adekwatnie, gdyż pliki można wciąż zapisywać na dysk, tylko nie w tym katalogu.

176 (Access denied) – pierwszą funkcją wywoływaną dla danego dysku przez *block_io* nie była funkcja nr 4 (*bio_rdsys*), podany numer dysku równy jest zero albo jest większy od 15, albo kod operacyjny funkcji jest nieprawidłowy.

179 „Memory conflict” – próba załadowania programu binarnego, który nakłada się na obszar kernela lub biblioteki SpartaDOS X. Często oznacza, że program musi być wywołany „przez X”.

181 „Filesystem corrupt” – DOS nie może wykonać żądanej operacji, bo struktura plików na dysku jest uszkodzona.