

**ATARI
MYDOS
4.50**

DYSKOWY SYSTEM OPERACYJNY
MyDOS wersja 4.50
dla 8-bitowych komputerów Atari

PODRĘCZNIK UŻYTKOWNIKA

Opracował Henryk Krasuski

Konwersja do formatu ODT oraz poprawki: Bluki, 06.05.2013

SPIS TREŚCI

I.	Wprowadzenie.....	4
II.	Wymagania sprzętowo - programowe.....	4
III.	Polecenia dostępne w menu MYDOS-a.....	5
	Nazwy plików i katalogów.....	5
IV.	Polecenia MYDOS-a.....	6
	Polecenie A - wyświetl zawartość katalogu lub informacje o określonych plikach.....	6
	Polecenie B - uruchom program z kartridża.....	7
	Polecenie C - kopiuj plik.....	7
	Polecenie D - kasuj plik.....	8
	Polecenie E - zmień nazwę pliku.....	8
	Polecenie F - nałóż blokadę zapisu na plik.....	8
	Polecenie G - znieś blokadę zapisu z pliku.....	8
	Polecenie H - zapisz system na dysk.....	9
	Polecenie I - formatuj dyskietkę.....	9
	Polecenie J - kopiuj dyskietkę.....	9
	Polecenie K - zapisz zawartość pamięci na dysk.....	10
	Polecenie L - wczytaj zawartość pliku do pamięci.....	10
	Polecenie M - skocz do podanego adresu.....	10
	Polecenie N - wczytaj zawartość pliku MEM.SAV do pamięci.....	10
	Polecenie O - definiowanie konfiguracji systemu i napędów dyskowych...11	
	Polecenie P - ustaw gęstość zapisu dyskietki.....	13
	Polecenie Q - utwórz podkatalog.....	14
	Polecenie R - określ katalog roboczy.....	14
V.	Funkcje zarządzania plikami realizowane przez procedurę CIO.....	14
VI.	Kody funkcji realizowanych przez procedurę CIO.....	15
	Funkcja 3 - przygotuj plik do operacji WE/WY.....	15
	Funkcja 5 - wczytaj rekord.....	16
	Funkcja 7 - wczytaj znaki.....	16
	Funkcja 9 - pisz rekord.....	16
	Funkcja 11 - pisz znaki.....	17
	Funkcja 12 - zamknij plik.....	17
	Funkcja 13 - czytaj status.....	17
	Funkcja 32 - zmień nazwę pliku.....	17
	Funkcja 33 - skasuj plik.....	17
	Funkcja 34 - utwórz podkatalog.....	18
	Funkcja 35 - nałóż blokadę zapisu na plik.....	18
	Funkcja 36 - zdejmij z pliku blokadę zapisu.....	18
	Funkcja 37 - ustaw wskaźnik pliku.....	18
	Funkcja 38 - odczytaj wskaźnik pliku.....	19
	Funkcja 39 - wczytaj program do pamięci.....	19
	Funkcja 41 - określ katalog roboczy.....	19
	Funkcja 254 - formatuj dyskietkę.....	19
VII.	Formaty dyskietek przetwarzane przez MYDOS-a.....	20
VIII.	Wykorzystanie pamięci przez MYDOS-a.....	20
IX.	Przystosowanie systemu do wymagań użytkownika.....	21
X.	Współpraca z napędem dyskowym.....	23
XI.	Współpraca z RAM-dyskiem.....	25
XII.	Instrukcja instalowania MYDOS-a.....	26
XIII.	Kody błędów.....	27
XIV.	VTOC Fixer - program kontroli tabeli zawartości wolumenu /VTOC/.....	29
	Od autora konwersji.....	32

I. WPROWADZENIE

Dyskowy system operacyjny przedstawiony w niniejszym podręczniku powstał później niż dobrze znany użytkownikom 8-bitowych Atari DOS 2.0. Jednakże można go traktować jako konsekwentne rozwinięcie koncepcji zastosowanej we wcześniejszych wersjach dyskowych systemów operacyjnych, czyniącej z MYDOS-a równie funkcjonalny i przyjazny dla użytkownika program. Dyskowe systemy operacyjne DOS 2.0 i DOS 2.5 dobrze współpracują z napędami dyskowymi Atari 810 i 1050, lecz mają znacznie ograniczone możliwości obsługi napędów charakteryzujących się większą gęstością zapisu informacji na dyskietce (np. XF-551) oraz dysków wirtualnych (RAM-dysków) o dużej pojemności.

II. WYMAGANIA SPRZĘTOWO - PROGRAMOWE

MYDOS wersja 4.50 ma cechy bardzo zbliżone do cech systemu DOS 2.0, to znaczy: zajmuje niewielki obszar pamięci komputera i umożliwia użytkownikowi wykorzystanie wszystkich komend (poleceń) dostępnych w DOS 2.0 z prawie identycznymi parametrami. Oznacza to, że MYDOS jest kompatybilny z wcześniejszymi wersjami dyskowych systemów operacyjnych, oferując jednocześnie wiele dodatkowych opcji.

Gdy MYDOS zostanie uruchomiony na komputerze wyposażonym w stację dysków umożliwiającą na dyskietce zapis jednostronny w pojedynczej gęstości, to będzie zapisywał i odczytywał pliki w formacie DOS 2.0 i DOS 2.5 (na dyskietce zostanie zapisanych 720 sektorów). Jednak niektóre programy mogą nie funkcjonować poprawnie lub w ogóle pod jego kontrolą. Przyczyną tego może być:

- niedozwolone odwołania do procedur DOS-a lub modułu FMS;
- przetwarzanie danych o rozmiarach katalogów i plików dyskowych (np. przetwarzanie trzycyfrowych wartości określających liczbę sektorów zajmowanych przez katalog lub plik).

MYDOS może być uruchomiony na 8-bitowym komputerze Atari dowolnego typu (XL, XE) z dołączoną co najmniej jedną stacją dysków elastycznych.

Gdy do komputera dołączona jest stacja dysków zezwalająca na zapis o zwiększonej gęstości, MYDOS używa własnego formatu plików. Format plików tworzonych w takiej sytuacji przez MYDOS-a nie jest kompatybilny ani z formatem stosowanym w systemie OS/A+ wersja 4, ani z formatem tworzonym przez SpartaDOS.

MYDOS może jednocześnie obsługiwać osiem napędów dysków i RAM-dysk, który jest wtedy traktowany jako napęd o numerze 9. Rezydentna część MYDOS-a (DOS.SYS) zawiera wszystkie funkcje realizowane przez DOS 2.0, tak więc można pod jego kontrolą uruchamiać większość z programów dedykowanych 8-bitowym Atari.

Obszar pamięci dostępnej dla programu uruchomionego pod kontrolą MYDOS-a zależy jedynie od liczby plików, które są jednocześnie przez niego otwarte. Każdy otwierany plik dyskowy wymaga istnienia związanego z nim 256 bajtowego bufora. W odróżnieniu od DOS 2.0 obszar pamięci dostępnej dla programu nie zależy ani od liczby dołączonych do systemu napędów dyskowych ani od rozmiaru sektora czy stosowanej gęstości zapisu. Rezerwowanie 256 bajtowego bufora dla każdego z jednocześnie otwartych plików oznacza, że programy przetwarzające w ten sposób wiele plików, będą dysponowały mniejszym obszarem wolnej pamięci, niż gdyby pracowały pod kontrolą DOS 2.0. W rezultacie dla niektórych z nich obszar dostępnej pamięci może okazać się zbyt mały. Dotyczy to przede wszystkim najstarszych wersji kompilatorów i interpreterów języków programowania. Przy zarezerwowaniu trzech 256 bajtowych buforów związanych z jednocześnie otwartymi plikami i przy braku rezydentnych sterowników programowych (na przykład łączy RS232) MYDOS zezwala na załadowanie programu już od adresu \$1F00.

MYDOS zawiera również bardzo funkcjonalny program obsługi RAM-dysku, którego objętość może dochodzić nawet do 1MB. Rozszerzenie pamięci, w której instalowany jest RAM-dysk musi odpowiadać standardowi rozszerzenia zastosowanego w komputerach Atari XE lub standardowi rozszerzenia Axlon dla komputerów Atari 800.

III. POLECENIA DOSTĘPNE W MENU MYDOS-a

W menu MYDOS-a jest dostępnych 18 poleceń. Polecenie jest wykonywane po naciśnięciu pojedynczego klawisza (litera), tak więc użytkownik nie musi pamiętać ani formatu polecenia ani potrzebnych do jego wykonania parametrów. Po wciśnięciu klawisza uaktywniającego określone polecenie MYDOS pyta o parametry potrzebne do jego wykonania (na przykład: nazwa pliku, na którym polecenie ma być wykonane, używana gęstość zapisu, itp.). Po wprowadzeniu potrzebnych parametrów MYDOS wykonuje polecenie i wyświetla znak zachęty informujący użytkownika o gotowości do dalszej pracy.

Użytkownika szczególnie zainteresują dwa elementy występujące w wyświetlanym menu. W drugiej linii ekranu są wyświetlane informacje charakteryzujące napędy dyskowe współpracujące z systemem (używana gęstość zapisu, RAM-dysk, dysk twardy). Następną linią określa napęd roboczy. Drugim elementem jest przewijanie menu w górę ekranu podczas wyświetlania informacji związanych z wykonywanym poleceniem. Dzięki temu na ekranie jest wyświetlanych więcej informacji o wykonywanym poleceniu lub sekwencji poleceń, co czasem bywa bardzo przydatne dla użytkownika. Menu zostaje ponownie wyświetlone po naciśnięciu klawisza RETURN.

Wykonywanie polecenia może być w dowolnej chwili przerwane przez naciśnięcie klawisza BREAK lub RESET. Gdy zdarzy się to podczas realizacji zapisu na dysk, to nie spowoduje to żadnych przykrych konsekwencji, aczkolwiek nazwa zapisywanego pliku prawdopodobnie nie pojawi się w katalogu dyskowym a wprowadzona liczba sektorów wolnych na dyskietce może być błędna.

NAZWY PLIKÓW I KATALOGÓW

Nazwa pliku może składać się maksymalnie z ośmiu znaków, którymi mogą być: wielkie i małe litery, cyfry, znak podkreślenia „_” oraz znak „@”. Pierwszy znak nazwy nie może być cyfrą. Ponadto nazwa może posiadać trzyznakowe rozszerzenie poprzedzone kropką. Nazwy plików wprowadzane jako parametry poleceń powinny być poprzedzone numerem napędu dyskowego i znakiem „:”. Gdy numer napędu nie zostanie wyspecyfikowany, polecenie zostanie wykonane na pliku znajdującym się w napędzie i katalogu roboczym. Gdy do polecenia [C]opy file /kopiuj plik/ zostanie wprowadzony jako jedyny parametr numer napędu, to w rezultacie zostaną skopiowane wszystkie pliki zapisane w katalogu głównym wyspecyfikowanego napędu. Specyfikacja napędu może być podana w jeden z następujących sposobów: „1”, „1” lub „D1:”. Specyfikacja ta - o ile zostanie wprowadzona - musi być zakończona znakiem „:”.

Przykłady: D1:TEST.CBJ, 1:TEST.ASM lub D2TEST (w rzeczywistości D1:D2TEST gdy napędem roboczym jest napęd o numerze 1) są nazwami poprawnymi, a d1:Test.obj lub 1TEST.ASM są nazwami błędnymi.

Nazwa pliku będąca parametrem polecenia może być podana w postaci tzw. wzorca. Do tworzenia wzorca służą znaki maskujące „*” i „?”.

Znak „*” użyty we wzorcu zastępuje kilka znaków w nazwie pliku, natomiast znak „?” zastępuje dokładnie jeden znak nazwy. Sposób interpretowania wzorca przez MYDOS-a ilustrują przedstawione niżej przykłady.

Wzorzec „*.BAS” obejmuje wszystkie nazwy plików mające rozszerzenie .BAS; wzorzec „TEST?.*” obejmuje wszystkie (maksymalnie pięciznakowe) nazwy plików, w których pierwszymi czterema znakami jest łańcuch „TEST” i które mają dowolne rozszerzenie. Nazwa TESTER.BAS nie spełnia wymagań określonych przez ten wzorzec.

Żałujemy, że w katalogu dyskowym zapisane są pliki o następujących nazwach: TEST.ASM, TEST.OBJ, TEST.C, TEST.ALM, TEASET.DOC, TRACE.FIL i BETS.LST.

Wzorzec „t*.*” nie obejmuje żadnej z nich, gdyż MYDOS rozróżnia małe i wielkie litery w nazwie pliku. Natomiast wzorzec „T*.*” obejmuje wszystkie przedstawione wyżej nazwy oprócz ostatniej (jej pierwszym znakiem jest „B” a nie „T”).

Wzorzec „?E??.*” obejmuje cztery pierwsze i ostatnią z powyższych nazw, gdyż ogranicza on długość nazwy do czterech znaków i wymaga, aby drugim znakiem nazwy była litera „E”.

Wzorzec „*.?” obejmuje jedynie nazwę TEST.C, gdyż ogranicza on długość rozszerzenia nazwy do jednego znaku. MYDOS nie zawsze wymaga, aby rozszerzenie nazwy pliku było poprzedzone kropką. Na przykład nazwa FILENAME.TXT może być podana w postaci FILENAMETXT, a wzorzec TEST*. * - w postaci TEST*. Jednak wprowadzenie wzorca TEST.* w postaci TEST* potraktowane zostanie przez MYDOS jako błąd.

Opisane wyżej zasady tworzenia nazw plików i ich wzorców dotyczą również nazw katalogów i podkatalogów dyskowych.

Katalog główny tworzony na dyskietce przez MYDOS-a może pomieścić nazwy 64 plików lub podkatalogów. Każdy utworzony podkatalog może również pomieścić nazwy 64 plików lub podkatalogów. Do rozdzielania nazw podkatalogów można użyć znaku „:” lub znaku „>” (znak większości). Na przykład: jeśli katalog główny zawiera podkatalog o nazwie BAS, w którym jest zapisany plik o nazwie GRAPHICS, to ścieżka dostępu do tego pliku może mieć jedną z dwóch postaci: BAS:GRAPHICS lub BAS>GRAPHICS.

Jeśli plik o wymienionej wyżej nazwie jest zapisany w podkatalogu GR.DIR utworzonym w podkatalogu BAS, to ścieżka dostępu do niego przyjmie postać BAS:GR.DIR:GRAPHICS. W tworzonych w opisany wyżej sposób ścieżkach dostępu można używać dowolną liczbę nazw podkatalogów. Wynika to z faktu, że MYDOS praktycznie nie ogranicza liczby poziomów zagnieżdżeń podkatalogów (w utworzonym podkatalogu można utworzyć następny, itd.).

Liczba poziomów zagnieżdżeń podkatalogów jest ograniczona jedynie przez liczbę wolnych sektorów i przez rozmiar bufora używanego przez program do przetwarzania podkatalogów. Zatem użytkownik na jednej dyskietce może przechowywać setki plików pogrupowanych w podkatalogi według własnego kryterium. Każdy podkatalog „jest widziany” przez podkatalog nadrzędny jako ośmio sektorowy plik.

IV. POLECENIA MYDOS-a

A. List a Directory or a Set of Files

/Wyświetl zawartość katalogu lub informacje o określonych plikach/

Polecenie A wyprowadza informacje o wszystkich (lub wskazanych przez wzorzec) plikach zapisanych w katalogu dyskowym. Wyprowadzane informacje zawierają nazwy plików i liczbę zajmowanych przez każdy plik sektorów. Na końcu jest wyprowadzana 4 lub 5 cyfrowa liczba wolnych sektorów na dysku. Gdy nazwę pliku poprzedza znak „*” oznacza to, że na plik ten została nałożona blokada zapisu (jego skasowanie lub modyfikacja zapisanych w nim informacji jest możliwa dopiero po wcześniejszym usunięciu blokady). Znak „:” poprzedza nazwy podkatalogów.

Wyprowadzane informacje o plikach nie zawierają danych wskazujących jaki format ma każdy z nich. Należy jednak pamiętać, że MYDOS może zapisywać i odczytywać pliki jedynie w trzech formatach: DOS-a 2.0, DOS-a 2.5 i własnym. Pliki mające format SpartaDOS lub systemu OS/A+ wersja 4 muszą być poddane konwersji, aby mogły być dostępne pod kontrolą MYDOS-a. Więcej szczegółów o informacjach wyprowadzanych przez polecenie A można znaleźć w rozdziale VI.

Polecenie A użyte z jednym parametrem (jest nim specyfikacja pliku) wyświetla informacje na ekran. Wprowadzenie drugiego parametru powoduje, że informacje o katalogu zostaną zapisane do pliku tekstowego określonego przez drugi parametr. Zapisanie informacji do takiego pliku może być wykonane na dwa sposoby. Pierwszy z nich niszczy starą zawartość wyspecyfikowanego pliku, drugi nie niszczy starej zawartości pliku, a nowe informacje są dopisywane na jej końcu. Drugi sposób wymaga użycia dodatkowego parametru /A. Na przykład: specyfikacja „2,D1:DIR.TXT” powoduje zapisanie informacji o zawartości głównego katalogu z napędu 2 do pliku tekstowego o nazwie DIR.TXT w napędzie 1. Specyfikacja „1,P:” powoduje wydrukowanie zawartości katalogu głównego z napędu 1.

Aby wyprowadzić informacje o zawartości podkatalogu, parametr polecenia A musi określać ścieżkę dostępu do niego. Na przykład: parametr „1:TEST:BAS:” powoduje, że zostaną wyświetlone na ekranie informacje o zawartości podkatalogu o nazwie BAS, zawartego w katalogu TEST w napędzie 1.

B. Run the Cartridge

/Uruchom program zapisany w dołączonym zewnętrznym module pamięci ROM/

W wyniku wykonania polecenia B kontrola nad komputerem jest przejmowana przez program zapisany w dołączonym zewnętrznym module pamięci ROM (kartridżu). Polecenie to nie wymaga określania żadnych parametrów. Gdy do komputera nie jest dołączony zewnętrzny moduł pamięci ROM, MYDOS sygnalizuje błąd. Gdy aktywna jest opcja ładowania do pamięci pliku MEM.SAV, to jego zawartość zostanie wczytana do pamięci przed uruchomieniem programu zawartego w kartridżu. Powyższa opcja jest uaktywniana poleceniem N.

C. Copy a File or a Set of Files

/Kopiuje plik lub zestaw plików/

Polecenie C służy do wykonania kopii jednego lub kilku plików. Wymaga ono określenia dwóch parametrów (specyfikacji plików). Pierwszy z nich określa plik lub pliki źródłowe tj. te, które mają być kopiowane. Drugi parametr definiuje kopie. Oba parametry mogą definiować również urządzenia. np. E: - monitor ekranowy, P: - drukarka, R1:, R2:, R3:, R4: - złącze RS232. Drugi parametr może być podany w postaci wzorca jedynie wtedy, gdy nazwa kopii może być w sposób jednoznaczny określona przez nazwę pliku kopiowanego.

Natomiast jeśli pierwszy parametr zostanie podany w postaci wzorca a drugi będzie definiował nazwę pliku i nie zostanie użyta opcja /A (patrz niżej), to wykonana kopia będzie zawierać informacje z ostatniego z plików źródłowych wskazanych przez podany wzorzec. Gdy w podanym parametrze zostanie pominięte oznaczenie katalogu i napędu dyskowego, to MYDOS założy, że określa on plik w roboczym napędzie i katalogu.

Podczas kopiowania plików wykorzystywany jest cały obszar pamięci. Dlatego każdy program, którego realizacja została zawieszona przed wykonaniem polecenia C nie da się restartować po jej zakończeniu. To samo dotyczy realizacji polecenia J - kopiuje dyskietkę.

Określając plik lub pliki do skopiowania można użyć jednej lub kilku z przedstawionych niżej opcji.

/A - umożliwia utworzenie z kilku plików źródłowych jednego pliku - kopii. Każdy plik źródłowy jest dopisywany do tworzonej kopii za ostatnim skopiowanym plikiem.

/Q - powoduje, że MYDOS przed skopiowaniem każdego z wyspecyfikowanych plików źródłowych żąda potwierdzenia, czy plik ten ma być skopiowany. Opcja ta jest użyteczna, gdy pliki źródłowe zostały określone przy pomocy wzorca (np. *.*), lecz nie wszystkie mają być skopiowane.

/S - eliminuje kopiowanie plików, których pierwszym znakiem rozszerzenia nazwy jest litera „S”. Jest ona przydatna, gdy trzeba skopiować wszystkie pliki z dyskietki oprócz tych, których nazwy mają rozszerzenie np. „SYS”.

/X - powoduje, że MYDOS będzie wyświetlał komunikat o potrzebie wymiany dyskietki w napędzie. Opcja ta jest przydatna, gdy podczas kopiowania używany ma być tylko jeden napęd. Jednak, gdy opcja ta zostanie użyta, to komunikat ten będzie wyświetlany nawet wtedy, gdy napęd zawierający pliki źródłowe jest różny od napędu, w którym są tworzone ich kopie.

Poniżej przedstawiono kilka przykładowych parametrów polecenia C oraz wyjaśniono ich wpływ na sposób jego wykonania.

TEST.BAS

Kopiuje plik o nazwie TEST.BAS zapisany w roboczym napędzie i katalogu. Kopia utworzona będzie w tym samym napędzie i katalogu. MYDOS „zapyta” o nazwę pliku - kopii, który ma utworzyć.

D1:TE*.*,3

Kopiuje z napędu o numerze 1 wszystkie pliki, których nazwy zaczynają się od znaków „TE”. Utworzone kopie będą miały takie same nazwy jak pliki źródłowe i zostaną zapisane w napędzie 3.

D4:DIR>*.*.TXT,E:/Q

Kopiuje wszystkie pliki z katalogu DIR z napędu 4, których nazwy mają rozszerzenie .TXT. Pliki są kopiowane na ekran. Przed skopiowaniem każdego pliku MYDOS żąda potwierdzenia, czy plik ten ma być skopiowany.

2:ADD,2:LIST/A

Dopisuje zawartość pliku o nazwie ADD z napędu 2 do pliku o nazwie LIST w tym samym napędzie.

D. Delete a File or a Set of Files

/Kasuj plik lub pliki/

Polecenie D umożliwia skasowanie z dysku jednego lub kilka plików. MYDOS przed skasowaniem każdego pliku wyświetla pytanie: Delete file [nazwa pliku]? /kasować plik [nazwa pliku]?/. Udzielenie odpowiedzi potwierdzającej powoduje skasowanie pliku. W poleceniu tym jest dostępna opcja /N. Jej użycie powoduje, że wyspecyfikowany (plik lub grupa plików) zostanie skasowany bez wyświetlenia przedstawianego wyżej pytania.

E. Rename a File or Set of Files

/Zmień nazwę pliku/

Polecenie E umożliwia zmianą nazwy pliku lub katalogu (podkatalogu) dyskowego. Wymaga ono określenia dwóch parametrów. Pierwszy z nich wskazuje plik lub katalog, którego nazwa ma być zmieniona, drugi definiuje nową nazwę pliku lub katalogu. Na przykład: parametry „D2:TEST>BASIC>NOTPNT.BAS,RANDIO.BAS” zmieniają nazwę pliku NOTPNT.BAS zapisanego w katalogu D2:TEST>BASIC>. Nową nazwą tego pliku będzie RANDIO.BAS. Aby zmienić nazwę podkatalogu BASIC na ABASIC należy użyć parametrów: „D2:TEST>BASIC,ABASIC”.

UWAGA! Nigdy nie należy podawać pierwszego parametru w postaci wzorca. Na przykład: użycie parametrów w postaci „*.*,TEST” spowoduje zmianę nazwy każdego pliku z napędu i katalogu roboczego na nazwę TEST. W rezultacie będzie dostępny tylko pierwszy z plików noszących nową nazwę.

F. Lock a File or Set of Files

/Nałóż blokadę zapisu na plik lub pliki/

Polecenie F służy do nakładania na plik lub grupę plików blokady zapisu. Plik z nałożoną blokadą nie może być skasowany, nie można zmienić jego nazwy ani zmodyfikować jego treści. Wykonanie powyższych operacji na takim pliku wymaga wcześniejszego zdjęcia nałożonej blokady zapisu (patrz polecenie G). Blokada zapisu może być nałożona również na katalog lub podkatalog. Wówczas do katalogu nie będzie można zapisać żadnego pliku, nie będzie można go skasować ani zmienić jego nazwy. Na pliki zawarte w takim katalogu blokada zapisu nie jest automatycznie nakładana. Przed nazwami plików, na które nałożona została blokada zapisu, polecenie A wyprowadza znak „*”. Polecenie to może być użyte z opcją /Q. Jej użycie powoduje, że MYDOS przed nałożeniem blokady na plik lub katalog wyświetla pytanie: „Lock [nazwa pliku lub katalogu]?”. Udzielenie odpowiedzi potwierdzającej („Y”) powoduje nałożenie blokady.

G. Unlock a File or Set of Files

/Znieś blokadę zapisu z pliku lub grupy plików/

Polecenie G znosi blokadę zapisu nałożoną na plik lub katalog. Zniesienie blokady może być wykonane również w programie przez wywołanie procedury CIO z odpowiednim kodem funkcji. W poleceniu tym można użyć opcji /Q, której działanie jest analogiczne jak w poleceniu F.

H. Write MYDOS to Disk

/Zapisz system na dysk/

Polecenie H służy do skopiowania na dysk plików systemowych MYDOS-a, tj. plików DOS.SYS i DUP.SYS. Plik DOS.SYS zawiera rezydującą stale w pamięci procedurę zarządzającą plikami, procedurę wczytującą do pamięci i zapisującą na dysk zawartość pliku MEM.SAV oraz procedurę ładującą do pamięci nierezydentną część MYDOS-a, zapisaną w pliku DUP.SYS. Plik ten zawiera standardowy program w kodzie maszynowym. Program ten po uruchomieniu wyświetla menu MYDOS-a. Żaden z plików DOS.SYS i DUP.SYS nie może być zastąpiony przez analogiczny plik z systemów DOS 2.0 i DOS 2.5. Oba powinny być traktowane jako nierozłączna całość. Nigdy też nie należy kopiować na dyskietkę tylko jednego z nich. Jedyny wyjątek od tej reguły może mieć miejsce tylko wtedy, gdy z góry można przewidzieć, że programy które zostaną dopisane na dyskietkę nie będą używały programu DUP.SYS (program ten jest uruchamiany np. z poziomu języka BASIC komendą DOS). Pliki DOS.SYS i DUP.SYS zapisane na dysk poleceniem H zawierać będą parametry konfiguracyjne systemu zdefiniowane w pamięci w chwili zapisywania plików.

I. Initialize a Diskette

/Formatuj dyskietkę/

Polecenie I realizuje proces formatowania dyskietki konieczny do jej wykorzystania przez MYDOS-a. Podczas formatowania na dyskietce oznaczane są ścieżki, zapisywane są informacje wykorzystywane przez system operacyjny i jest tworzony pusty katalog główny. Parametrem polecenia jest numer napędu dyskowego, w którym jest dyskietka do sformatowania. Jeżeli numer napędu dyskowego zostanie podany z opcją /N, to dyskietka nie jest formatowana lecz są kasowane na niej wszystkie podkatalogi z zapisanymi w nich plikami i jest wpisywany pusty katalog główny. Opcja /N udostępnia najszybszy sposób skasowania z dyskietki wszystkich zapisanych na niej informacji. Należy zaznaczyć, że opcji tej można użyć jedynie wtedy, gdy istnieje pewność, że dyskietka nie zawiera błędnych sektorów. Polecenie I umożliwia również sformatowanie dyskietki w zwiększonej gęstości zapisu możliwej do uzyskania w napędzie dyskowym Atari 1050 (lub kompatybilnym). Po naciśnięciu klawisza I MYDOS wyświetla pytanie: „Type [A] to format in Enhanced. Type [Y] to format drive [numer napędu]?”. Gdy w odpowiedzi zostanie naciśnięty klawisz „A” i napęd Atari 1050 lub kompatybilny jest dołączony do komputera, to dyskietka zostanie w nim sformatowana ze zwiększoną gęstością zapisu. Należy zaznaczyć, że tak sformatowanej dyskietki nie da się odczytać pod kontrolą systemów DOS 2.0 i DOS 2.5. Natomiast dyskietki sformatowane przez wymienione systemy będą odczytywane przez MYDOS-a. Przed rozpoczęciem formatowania dobrze jest zdefiniować gęstość zapisu, która ma być użyta do formatowania. Jest to istotne szczególnie wtedy, gdy napęd, w którym znajduje formatowana dyskietka nie zezwala na zapis o podwójnej gęstości. Definiowanie gęstości zapisu przedstawione zostało w opisie polecenia P. Gdy gęstość nie zostanie zdefiniowana przed formatowaniem, to MYDOS zastosuje gęstość użytą do ostatnio zrealizowanego procesu formatowania.

J. Duplicate a Diskette

/Kopiuje dyskietkę/

Polecenie J kopiuje wszystkie informacje zapisane na wskazanej dyskietce na inną dyskietkę. Można skopiować również fragment dyskietki przez wskazanie sektorów, które mają być skopiowane (sektor początkowy - sektor końcowy). Gdy nie zostaną wskazane sektory do skopiowania, MYDOS kopiuje całą dyskietkę, przepisując wszystkie te sektory, które w tabeli zawartości wolumenu /VTOC/ oznaczone są jako zajęte. Sektor zawierający VTOC może być skonstruowany przez Atari DOS 2.0, dowolną wersją MYDOS-a lub każdy inny system operacyjny kompatybilny z wyżej wymienionymi. Na przykład, aby skopiować sektory od 19 do 54 z dyskietki znajdującej się w napędzie 1 na dyskietkę w napędzie 3 z pominięciem formatowania dyskietki, na której będzie tworzona kopia, należy użyć polecenia J z parametrami: „1,3/N(19-54)”.

W celu skopiowania zawartości całej dyskietki z napędu 1 na dyskietkę w napędzie 2 z formatowaniem dyskietki kopii, należy wprowadzić parametr: „1,2(1-720)” lub „1,2(1-1040)”. Przykłady te ilustrują sposób wykonania kopii dyskietek nie mających formatu MYDOS-a. Gdy dyskietka, na której jest tworzona kopia, została wcześniej poprawnie sformatowana, opcji /N można użyć albo po specyfikacji numeru napędu zawierającego dyskietkę źródłową albo po specyfikacji napędu zawierającego tworzoną kopię. Oznacza to, że parametry „1/N,2” oraz „1,2/N” dają ten sam rezultat, tj. skopiowanie dyskietki z napędu 1 na dyskietkę w napędzie 2 bez formatowania tej ostatniej. Pominięcie opcji /N powoduje, że MYDOS przed rozpoczęciem kopiowania sformatuje dyskietkę wyspecyfikowaną jako kopia. Podczas kopiowania dyskietki wykorzystywany jest cały obszar pamięci komputera. W rezultacie każdy program, którego wykonywanie zostało zawieszony przed kopiowaniem, nie da się restartować po jego zakończeniu.

K. Save Memory to Disk

/Zapisz zawartość pamięci na dysk/

Polecenie K umożliwia zapisanie na dysk danych rezydujących w wyspecyfikowanym obszarze pamięci. Jeśli dane te tworzą program (kod maszynowy), to może on być uzupełniony o wektor inicjacji (INIT adres) i wektor startu (RUN adres). Gdy dane zapisywane na dysk nie są programem, to oba adresy (inicjacji i startu) powinny być pominięte lub podane jako zera. Adresy definiujące początek i koniec obszaru pamięci, zawartość którego ma być zapisana na dysk, adres inicjacji i adres startu powinny być podawane w postaci liczb szesnastkowych. W poleceniu K można użyć opcji /A, umożliwiającej dopisanie zawartości pamięci do już istniejącego pliku. Polecenia K można użyć również do dopisania np. wektora startu do istniejącego pliku zawierającego program. Na przykład: parametr „D1:PROGRAM.OBJ/A,0,0,,4000” powoduje dopisanie wektora startu (start od adresu \$4000) do programu zapisanego w pliku PROGRAM.OBJ w napędzie 1. Gdy aktywna jest opcja ładowania do pamięci pliku MEM.SAV, to przed wykonaniem polecenia K zawartość tego pliku zostanie wczytana do pamięci.

L. Load Memory from a File /Disable MEM.SAV/

/Wprowadź zawartość pliku do pamięci - zablokuj opcję ładowania pliku MEM.SAV do pamięci/

Polecenie L umożliwia załadowanie do pamięci zawartości pliku binarnego (programu w kodzie maszynowym). Jeśli w pliku tym są wyspecyfikowane wektory startu i inicjacji, to w czasie ładowania będą uruchamiane wszystkie procedury inicjujące i po zakończeniu ładowania program zostanie automatycznie uruchomiony. Jeśli po nazwie pliku zostanie użyta opcja /N, to zapisane w pliku wektory inicjacji i startu będą ignorowane (program po załadowaniu nie zostanie automatycznie uruchomiony). Jeśli polecenie L zostanie użyte do załadowania pliku nie będącego programem w kodzie maszynowym, to MYDOS zasygnalizuje błąd nr 180, gdy ładowany plik binarny jest uszkodzony - błąd nr 181.

Polecenie L bez parametru (naciśnięcie „L” i RETURN) blokuje opcję ładowania do pamięci zawartości pliku MEM.SAV. Gdy opcja ta jest aktywna, to plik MEM.SAV jest automatycznie wczytywany przez załadowaniem i uruchomieniem dowolnego programu.

M. Run at Address

/Skocz do podanego adresu/

Polecenie M umożliwia uruchomienie programu od wyspecyfikowanego adresu, gdy program ten po załadowaniu do pamięci nie jest automatycznie uruchamiany. Można go użyć również do restartowania pracy komputera przez podanie adresu \$E477 (adres skoku do procedury „zimnego startu”).

N. Load MEM.SAV from a File /Enable MEM.SAV/

/Wprowadź zawartość pliku do pamięci - uaktywnij opcję ładowania do pamięci pliku MEM.SAV/

Gdy polecenie to zostanie użyte z parametrem - nazwą pliku, to jego zawartość zostanie załadowana do pamięci komputera. W takim przypadku, polecenie N realizowane jest tak samo jak polecenie L.

Różnica między nimi tkwi w tym, że polecenie N może być użyte do uaktywnienia opcji ładowania do pamięci zawartości pliku MEM.SAV. Gdy opcja ta jest aktywna, to przed załadowaniem i uruchomieniem dowolnego programu, jest wczytywana do pamięci zawartość pliku MEM.SAV. Następnie, gdy program zostanie wykonany, to przed przekazaniem kontroli nad komputerem ponownie do MYDOS-a, zawartość pamięci jest zapisywana do pliku MEM.SAV. W celu uaktywnienia tej opcji należy użyć polecenia N bez parametru, tj. nacisnąć „N” i RETURN.

O. System and Drive Configuration

/Definiowanie konfiguracji systemu i napędów dyskowych/

Polecenie O umożliwia określenie parametrów definiujących konfigurację systemu komputerowego i zdefiniowanie dołączonych do niego napędów dyskowych. Parametrami definiującymi konfigurację systemu są: liczba buforów pamięci zarezerwowanych dla jednocześnie otwartych plików, obecność RAM-dysku, realizacja zapisu na dysk z weryfikacją lub bez, itp. Definiowanie parametrów konfiguracyjnych systemu, nie związanych z dołączonymi napędami dyskowymi jest realizowane przez naciśnięcie RETURN po wyświetleniu następującego komunikatu: „Drive number or RETURN?”. Proces definiowania polega na udzieleniu odpowiedzi na wyświetlane przez MYDOS-a pytania. Są nimi:

1. Verify WRITES /Czy realizować zapis z weryfikacją?/

Naciśnięcie klawisza „N” oznacza, że zapis na dysk będzie realizowany bez weryfikacji. Naciśnięcie każdego innego klawisza oznacza realizowanie zapisu z weryfikacją.

2. Number of File Buffers? /Liczba buforów/

Należy wprowadzić liczbę buforów, które będą zarezerwowane dla jednocześnie otwartych plików i nacisnąć RETURN. Naciśnięcie samego klawisza RETURN powoduje, że MYDOS przyjmuje, iż liczba takich buforów jest równa 3.

3. RAM disk present? /Czy będzie instalowany w systemie RAM-dysk?/

Naciśnięcie „N” oznacza, że w systemie nie ma RAM-dysku. W takiej sytuacji cztery następne pytania nie zostaną wyświetlone. Naciśnięcie dowolnego innego klawisza stanowi dla MYDOS-a informację, że w systemie obecny jest RAM-dysk i odpowiedzi udzielone na następne pytania umożliwiają określenie jakiego jest on rodzaju.

4. [A]xlon or [X]E type? /Typ RAM-dysku: Axlon lub XE?/

Naciśnięcie „A” oznacza, że RAM-dysk jest typu Axlon. naciśnięcie „X” oznacza, że RAM-dysk ma typ, który zastosowano w komputerach Atari XE. Gdy w odpowiedzi naciśnięty zostanie klawisz „A”, następne pytanie zostanie pominięte.

5. Use default config for (x)K?

/Czy użyć standardowych parametrów do utworzenia RAM-dysku?/

Naciśnięcie „N” oznacza, że parametry definiujące RAM-dysk zostaną podane przez użytkownika. Naciśnięcie „Y” lub RETURN oznacza, że MYDOS ma użyć standardowych parametrów do instalowania RAM-dysku. W wyświetlonym pytaniu (x) oznacza domniemaną przez MYDOS-a wielkość obszaru pamięci (w kilobajtach), która będzie wykorzystana jako RAM-dysk. Gdy udzielona zostanie odpowiedź „Y”, następne trzy pytania zostaną pominięte.

6. Size (K)? /Rozmiar RAM-dysku w kilobajtach?/

Wprowadzony rozmiar RAM-dysku musi być liczbą podzielna przez 16. Dla rozszerzenia pamięci NI 256K (NEWELL INDUSTRIES) należy wprowadzić liczbę 192 lub 128, dla rozszerzenia typu Axlon - liczbę 112, dla komputera Atari 130XE - liczbę 64.

7. Control Address (HEX)? /Adres sterowania dostępem do rozszerzenia pamięci?/

Parametr ten definiuje adres rejestru (bajtu pamięci), którego zawartość jest wykorzystywana w sterowaniu dostępem do rozszerzenia pamięci. Dla komputera Atari 130XE adres ten jest równy \$D301 (aby go wprowadzić wystarczy tylko nacisnąć RETURN). Dla rozszerzenia pamięci AXLON RAMPOWER 128 należy wprowadzić adres \$CFFF.

8. Page sequence? /Tablica sekwencji stron?/

Naciśnięcie klawisza RETURN oznacza, że RAM-dysk będzie instalowany w rozszerzeniu pamięci zgodnym ze standardem zastosowanym w komputerach Atari 130XE. Gdy RAM-dysk będzie tworzony w rozszerzeniu NEWELL INDUSTRIES, należy wprowadzić wartość 1, dla rozszerzenia RAMBO ICD - wartość 2, dla rozszerzenia AXLON i rozszerzeń z nim kompatybilnych - wartość 5. Ogólna zasada definiowania tablicy sekwencji stron jest następująca: należy wprowadzić ciąg dwucyfrowych liczb szesnastkowych reprezentujących numery stron pamięci, oddzielając je przecinkami. Ciąg ten należy zakończyć wartością \$00, gdy rozszerzenie pamięci ma typ zgodny z rozszerzeniem zastosowanym w Atari 130XE, lub wartością \$FF, gdy rozszerzenie jest typu Axlon.

9. RAM disk drive no? /Numer napędu, z którym zostanie skojarzony RAM-dysk?/

Należy wprowadzić numer napędu (z zakresu 1 - 9), z którym skojarzony będzie RAM-dysk. Naciśnięcie RETURN kojarzy RAM-dysk z napędem nr 9.

Jeśli opisany wyżej proces konfigurowania systemu zostanie z jakiegokolwiek przyczyny przerwany, to zdefiniowane w nim parametry nie będą miały wpływu na konfigurację. Aby wszystkie wprowadzone parametry zostały zaakceptowane przez system, proces ich definiowania musi zostać zakończony poprawnie.

DEFINIOWANIE NAPĘDU DYSKOWEGO

Jeśli po wyświetleniu komunikatu „Drive number or RETURN” zostanie wprowadzony numer napędu dyskowego, to rozpocznie się proces jego definiowania. Proces ten, podobnie jak opisany wyżej, polega na udzielaniu odpowiedzi na wyświetlane przez MYDOS-a pytania. W odpowiedzi na pierwsze pytanie należy określić, czy obecność definiowanego napędu w systemie ma być sprawdzana podczas procesu inicjacji (włączenie komputera lub RESET). Udzielenie odpowiedzi przeczącej kończy proces definiowania napędu. Odpowiedź potwierdzająca powoduje wyświetlenie następnego pytania: „Czy możliwa jest konfiguracja napędu?”. Niektóre napędy dyskowe, jak Atari 810 nie pozwalają na zmianę swych parametrów. Zatem konfiguracja takich napędów nie jest możliwa. Odpowiedź „N” (nie) powoduje, że MYDOS zakłada, że definiowany napęd jest kompatybilny z napędami Atari 810/815/1050. Na dyskietce formatowanej w napędach tej klasy MYDOS będzie zapisywał 720 sektorów.

Gdy konfiguracja napędu jest możliwa (można modyfikować jego parametry), wyświetlane jest trzecie pytanie: „Czy jest to napęd o dużej pojemności (high capacity drive)?”. Innymi słowy: MYDOS pyta czy jest to napęd dysków elastycznych, czy napęd dysku twardego. Naciśnięcie klawisza „N” oznacza, że definiowany napęd jest napędem dysków elastycznych, a klawisza „Y” wskazuje, że jest definiowany napęd dysku twardego. W drugim przypadku MYDOS zażąda podania liczby sektorów, które można na nim zapisać. W odpowiedzi należy wprowadzić liczbę z zakresu 256 - 65535). MYDOS może współpracować z dyskami twardymi o pojemności do 16MB. Dyski o większej pojemności należy podzielić na partycje o pojemności nie przekraczającej 16 MB i zdefiniować każdą partycję jako oddzielny napęd. Więcej informacji na ten temat można znaleźć w dokumentacji używanego napędu dysku twardego.

Jeśli zdefiniowany napęd będzie wyłączony (lub odłączony od komputera) podczas włączania komputera, to mimo to system zostanie załadowany poprawnie. Możliwa jest również sytuacja odwrotna, tj. MYDOS będzie współpracował poprawnie z napędem, który nie został w systemie zdefiniowany a jest podłączony do komputera podczas ładowania systemu operacyjnego. Definiowanie napędu dyskowego w systemie jest konieczne jedynie w dwóch przypadkach: gdy użyty ma być napęd dysku twardego lub gdy jest używany interfejs PERCOM lub ATR8000. W każdym innym przypadku definiowanie napędu nie jest konieczne.

Jeśli na przedstawione wyżej trzy pytania udzielone zostały następujące odpowiedzi:

„N” (MYDOS podczas włączania lub restartu komputera ma sprawdzać obecność napędu w systemie),
 „Y” (modyfikacja parametrów napędu jest możliwa) i „N” (jest to napęd dysków elastycznych),

to MYDOS dodatkowo zażąda udzielenia odpowiedzi na następujące pytania:

1. Czy napęd umożliwia zapis dwustronny, czy jednostronny?
2. Ile ścieżek można sformatować na każdej stronie dyskietki?
3. Jaki jest czas przesuwu głowic pomiędzy sąsiednimi ścieżkami?

Odpowiadając na pierwsze pytanie można nacisnąć klawisz „N” lub „Y”. Odpowiedź „Y” oznacza, że napęd jest dwustronny i dyskietki będą zapisywane w nim dwustronnie. Odpowiadając na drugie pytanie należy wprowadzić jedną z następujących liczb: 35, 40, 77 lub 80 i nacisnąć RETURN. Na przykład: jeśli definiowany jest napęd umożliwiający dużą gęstość zapisu (stosowany w PC IBM/AT o pojemności 1,2 MB), to można wprowadzić liczbę ścieżek równą 77. W tak zdefiniowanym napędzie MYDOS będzie zapisywał na dyskietce około miliona znaków. Liczba wprowadzona w odpowiedzi na drugie pytanie zawiera również informacje o typie napędu (napęd 8. calowy, standardowy napęd 5 i 1/4 cala lub napęd 5 i 1/4 cala o dużej pojemności).

Odpowiedź na ostatnie – trzecie pytanie musi być podana w postaci kodu. Przedstawiona niżej tabelka zawiera dopuszczalne kody i odpowiadające im czasy przesuwu głowic pomiędzy sąsiednimi ścieżkami dla różnych typów napędów.

KOD	CZAS PRZESUWU GŁOWIC MIĘDZY SĄSIEDNIMI ŚCIEŻKAMI	
	Napęd 8 calowy	Napęd 5 i 1/4 calowy
0	3 ms	6 ms /napęd XF-551/
1	6 ms	12 ms
2	10 ms	20 ms
3	15 ms	30 ms

P. Diskette Density Selection

/Ustaw gęstość zapisu dyskietki/

MYDOS realizując większość poleceń jest w stanie sam określić, w jakiej gęstości są zapisane przetwarzane dyskietki. W związku z tym użytkownik nie musi martwić się tym problemem. Jednak w menu MYDOS-a dostępne jest polecenie P, umożliwiające użytkownikowi określenie gęstości zapisu. Służy ono przede wszystkim do „wymuszenia” gęstości zapisu, jakiej MYDOS ma użyć przy formatowaniu dyskietki. Polecenie to jest przydatne również wtedy, gdy MYDOS nie jest w stanie automatycznie rozpoznać z jaką gęstością zapisane są informacje na dyskietce. MYDOS rozpoznaje gęstość zapisu odczytując pierwszy sektor. W niektórych napędach dyskowych, jak INDUS GT, MYDOS nie jest w stanie rozpoznać gęstości zapisu i wówczas należy użyć polecenia P.

Aby ustawić gęstość zapisu dla określonego napędu, należy podać jego numer i literę „S” (pojedyncza gęstość) lub „D” (podwójna gęstość) poprzedzone przecinkiem. Na przykład: „2,D”.

Q. Create Subdirectories

/Utwórz podkatalog/

Gdy MYDOS formatuje dyskietkę, to tworzy i zapisuje na niej pusty katalog główny. W katalogu tym można zapisać nazwy 64 plików lub podkatalogów. Podkatalog może być utworzony w istniejącym katalogu głównym lub podkatalogu, który będzie podkatalogiem nadrzędnym w stosunku do tworzonego podkatalogu.

Aby utworzyć podkatalog należy użyć polecenia Q. Parametrem polecenia jest nazwa tworzonego podkatalogu. Na przykład: jeśli w katalogu głównym napędu 1 istnieją dwa podkatalogi TEST i BAS, to polecenie Q z parametrem „D1:TEST:COMM” tworzy podkatalog o nazwie COMM w podkatalogu TEST, a z parametrem „1:BAS>COMM” tworzy podkatalog COMM w podkatalogu BAS. Natomiast użycie polecenia Q z parametrem „D1:NEW:COMM” nie utworzy żadnego podkatalogu, gdyż w napędzie 1 nie ma podkatalogu o nazwie NEW, Aby utworzyć taki podkatalog, należy użyć polecenia Q z parametrem „1:NEW” i dopiero po jego wykonaniu można użyć parametru „D1:NEW:COMM” do utworzenia podkatalogu COMM.

Każdy podkatalog zajmuje 8 sektorów dyskietki. Utworzony podkatalog można skasować jedynie wtedy, gdy nie ma zapisanych w nim żadnych plików. Do skasowania plików z podkatalogu należy użyć polecenia D.

R. Set the Default Directory

/Określ katalog roboczy/

Podkatalogiem lub katalogiem roboczym jest podkatalog przeszukiwany przez MYDOS-a, gdy w poleceniu zostanie użyta nazwa pliku bez specyfikacji numeru napędu, np. „TEST1.BAS”, „D:PROGRAM.BAS” lub „:PROGRAM.C”. Polecenie R umożliwia określenie, który z podkatalogów ma być przeszukiwany w takich sytuacjach. Katalog roboczy można również określić w programie przez wywołanie procedury CIO z kodem funkcji 41. Przed określeniem katalogu roboczego należy do odpowiedniego napędu włożyć dyskietkę, na której jest on zapisany, a następnie użyć polecenia R z parametrem specyfikującym numer napędu i nazwę katalogu. Za nazwą katalogu nie należy umieszczać znaku „:” ani „>”. Na przykład: jeśli katalogiem roboczym ma być podkatalog o nazwie BASIC, zapisany w podkatalogu TEST i dyskietka zawierająca go znajduje się w napędzie o numerze 5, to należy użyć polecenia R z parametrem „D5:TEST>BASIC”. Jeśli dyskietka zawierająca katalog określony jako roboczy zostanie usunięta z napędu dyskowego lub zostanie z niej skasowany katalog roboczy, to należy ponownie określić taki katalog.

Plik MEM.SAV

Zawsze, gdy jest uaktywniane menu MYDOS-a, do pamięci komputera jest ładowany program zapisany w pliku DUP.SYS. MYDOS szuka tego pliku w napędzie 1, jeśli go tam nie znajdzie, szuka w napędzie 2, 3, itd., aż plik ten zostanie znaleziony. Załadowany program wykorzystuje początkowe strony pamięci komputera i jeśli przed uaktywnieniem menu MYDOS-a były używane przez inny program to zapisane w nich wcześniej dane zostaną zniszczone. Problemu tego można uniknąć przez użycie pliku o nazwie MEM.SAV. W menu MYDOS-a dostępne jest polecenie N, uaktywniające opcję wczytywania do pamięci zawartości pliku MEM.SAV. Gdy opcja ta jest aktywna, wówczas przed załadowaniem do pamięci programu z pliku DUP.SYS, jej zawartość zapisywana jest do pliku MEM.SAV w napędzie 1 (lub 8, gdy używany jest jeden z programów RAMBOOT z dyskietki systemowej). Gdy następuje wyjście z uruchomionego programu, zawartość pamięci jest odtwarzana z pliku MEM.SAV. W rezultacie program rezydujący w pamięci komputera przed uaktywnieniem menu MYDOS-a zostanie do niej ponownie załadowany i można go bez przeszkód uruchomić.

V. FUNKCJE ZARZĄDZANIA PLIKAMI REALIZOWANE PRZEZ PROCEDURĘ CIO

MYDOS oferuje użytkownikowi takie same funkcje zarządzania plikami realizowane przez procedurę CIO, jakie były dostępne w Atari DOS 2.0. Niektóre z nich, jak: OPEN (kod funkcji 3) lub FORMAT (kod funkcji 254) zostały w nieznacznym stopniu zmodyfikowane. Inne, jak MAKE DIRECTORY (kod funkcji 34 i 42), SET DIRECTORY

(kod funkcji 41) i LOAD MEMORY (kod funkcji 39 i 40) są zupełnie nowymi funkcjami, które nie były dostępne w starszych wersjach dyskowego systemu operacyjnego.

Funkcja OPEN (PRZYGOTUJ URZĄDZENIE/PLIK DO OPERACJI WE/WY) realizowana przez procedurę CIO w DOS 2.0 nie wykorzystywała parametrów przekazywanych przez drugi bajt (AUX2) pomocniczy z bloku IOCB, natomiast realizowana przez MYDOS - wykorzystuje. Gdy bajt AUX1 zawiera wartość 8 (tworzenie pliku lub wymiana jego zawartości), to dwa bity bajtu AUX2 określają format pliku i czy na plik ma być nałożona blokada zapisu, czy nie. Jeśli ustawiony jest drugi bit bajtu AUX2, to tworzony plik będzie miał format MYDOS-a. W rezultacie może on zawierać sektory o numerze większym od 1023. Pliki mające taki format nie dadzą się w prosty sposób odczytać pod kontrolą systemów innych niż MYDOS. Format ten jest jedynym formatem, jakiego MYDOS używa, zapisując plik w napędzie o dużej pojemności (1.2 MB lub dysk twardy). Gdy ustawiony jest piąty bit bajtu AUX2, to na plik zostanie nałożona blokada zapisu. Opcja ta została opracowana z myślą o wykorzystaniu w przetwarzaniu wielozadaniowym, np. wyprowadzanie danych z kilku komputerów na jedną drukarkę, które można zaimplementować do standardowych zadań realizowanych przez system operacyjny i DOS.

Funkcja FORMAT (FORMATUJ DYSKIETKĘ) realizowana była zawsze w standardowy sposób przez Atari DOS 2.0. W systemie MYDOS sposób jej wykonania zależy od zawartości bajtów AUX1 i AUX2. Określa ona liczbę ścieżek na formatowanej dyskietce, czy dyskietka ma być fizycznie formatowana, czy nie i czy wymaga ona zainicjowania pustego katalogu głównego. Gdy ustawiony jest bit 7 bajtu AUX1, to dyskietka nie jest fizycznie formatowana (o ile nie jest to konieczne), lecz kasowane są wszystkie zapisane na niej informacje. Bity 0-6 bajtu AUX1 i wszystkie bity bajtu AUX2 określają liczbę sektorów na formatowanej dyskietce. Gdy wszystkie bity są zerami, to MYDOS zakłada, że pojemność dyskietki została określona w zdefiniowanej konfiguracji napędu dyskowego. Dzięki temu możliwe jest np. sformatowanie dyskietki jednostronnie w napędzie dwustronnym, przy czym zalecana jest ostrożność przy korzystaniu z tej możliwości.

W celu wczytania do pamięci i uruchomienia programu można w MYDOS-ie użyć funkcji LOAD MEMORY (kod funkcji 39). Funkcja o kodzie 40 realizuje to samo zadanie i została opracowana z myślą o utrzymaniu kompatybilności z systemem SpartaDOS. Dzięki niej istnieje możliwość ładowania do pamięci i uruchomienia programów z poziomu języka BASIC za pomocą instrukcji XIO,39,#IOCB,4,0,"D:PROGRAM.OBJ" (IOCB jest dowolnym nieaktywnym blokiem kontroli WE/WY). Jeśli bajt AUX1 zawiera wartość 5 i wczytywany program ma określone wektory inicjacji i startu, to zostanie on automatycznie uruchomiony, jeśli AUX1 zawiera wartość 6, to zostaną wykonane jedynie procedury inicjujące, a wpisanie do AUX1 wartości 7 spowoduje, że ani procedury inicjujące nie zostaną wykonane, ani program nie zostanie uruchomiony.

Do utworzenia podkatalogu można użyć wywołania funkcji o kodzie 34. Funkcja o kodzie 42 realizuje to samo zadanie i również została zaimplementowana w celu utrzymania kompatybilności z systemem SpartaDOS. Utworzenie podkatalogu z poziomu języka BASIC jest realizowane przy pomocy instrukcji: XIO,34,#IOCB,8,0,"nazwa podkatalogu". Nazwa tworzonego podkatalogu musi być unikatowa, tzn. w podkatalogu nadrzędnym nie może istnieć nazwa pliku lub podkatalogu identyczna z podaną. Nazwa podkatalogu nie może być zakończona znakiem „:" i „>”.

Ostatnią z funkcji dodanych do MYDOS-a jest funkcja o kodzie 41. Służy ona do określenia podkatalogu roboczego, tj. podkatalogu, który będzie przeszukiwany, gdy w nazwie pliku nie zostanie wyspecyfikowany numer napędu dyskowego. Katalogiem roboczym w DOS 2.0 był zawsze główny katalog z napędu numer 1. W MYDOS-ie katalogiem roboczym może być katalog główny lub dowolny podkatalog z dowolnego napędu dyskowego.

VI. KODY FUNKCJI REALIZOWANYCH PRZEZ PROCEDURĘ CIO

KOD FUNKCJI 3, OPEN

REALIZOWANE ZADANIE: Przygotuj urządzenie/plik do operacji WE/WY.

Funkcja OPEN wymaga parametru - adresu bufora, zawierającego łańcuch znaków ATASCII. Łańcuch ten może być zakończony znakiem niealfanumerycznym, znakiem „*” lub „?”. Łańcuch reprezentuje nazwę pliku, który zostanie otwarty w wyniku jej wykonania. Dobrą praktyką jest umieszczanie na końcu łańcucha znaku „null” (\$00) lub znaku końca linii (RETURN - \$9B).

Wartość bajtu AUX1 bloku IOCB definiuje tryb otwarcia. Dozwolone są następujące tryby otwarcia:

Wartość AUX1	Tryb otwarcia
4	Otwarcie pliku do odczytu.
6	Otwarcie dla odczytu zawartości katalogu lub podkatalogu.
8	Otwarcie pliku do zapisu (w celu jego utworzenia lub zmiany jego zawartości).
9	Otwarcie pliku do zapisu (w celu jego utworzenia lub dopisania informacji na jego końcu).
12	Otwarcie pliku do odczytu i zapisu (aktualizacji).

Wartość bajtu AUX2 jest istotna, gdy plik został otworzony w trybie 8. W takim przypadku, gdy ustawiony jest bit 2 bajtu AUX2, to tworzonemu plikowi zostanie nadany format MYDOS-a nawet jeśli dyskietka, na którą zapisywany jest plik, jest jednostronna i zawiera tylko 40 ścieżek na stronie. Ustawienie bitu 6 bajtu AUX2 powoduje, że na utworzony plik zostanie automatycznie nałożona blokada zapisu. Dla innych trybów otwarcia wartość bajtu AUX2 jest ignorowana.

KOD FUNKCJI 5, GET RECORD
 REALIZOWANE ZADANIE: wczytaj rekord.

Funkcja GET RECORD wczytuje rekord danych z pliku do bufora pamięci. Bufor jest określony przez adres początkowy (adres pierwszego bajtu bufora) i długość (pojemność bufora w bajtach).

Wczytany rekord może być niepełny, gdy jego długość przekracza pojemność bufora. Koniec rekordu w buforze jest sygnalizowany przez znak końca linii (\$9B). Wszystkie wykonywane przez MYDOS-a operacje wejścia/wyjścia, które działają na rekordach, są buforowane.

Oprócz przedstawionych wyżej dwóch parametrów funkcja GET RECORD nie wymaga przekazania do wykorzystywanego bloku IOCB żadnych innych argumentów. Jeśli długość bufora jest równa 0, to funkcja ta jest realizowana w sposób identyczny jak funkcja GET CHARACTERS.

KOD FUNKCJI 7, GET CHARACTERS
 REALIZOWANE ZADANIE: Wczytaj znak lub znaki.

Funkcja GET CHARACTERS odczytuje z pliku określoną liczbę znaków i przesyła je do bufora pamięci. Bufor, podobnie jak w funkcji o kodzie 5, jest zdefiniowany przez adres początkowy i długość. Funkcja nie prześle do bufora żadnego znaku, gdy zostanie wczytany znak końca pliku lub podczas operacji odczytu wystąpi błąd. Jeśli długość bufora jest równa 0, to funkcja odczyta jeden znak, który zostanie przesłany do akumulatora (rejestr A). Gdy długość bufora przekracza 256 bajtów, to operacja odczytu znaków nie jest buforowana.

Oprócz adresu początkowego i długości bufora funkcja GET CHARACTERS nie wymaga przekazania do wykorzystywanego bloku IOCB żadnych dodatkowych argumentów.

KOD FUNKCJI 9. PUT RECORD
 REALIZOWANE ZADANIE: Pisz rekord.

Funkcja PUT RECORD zapisuje pojedynczy rekord do pliku wyjściowego. Parametrami wykorzystywanymi przez funkcję są: adres bufora pamięci zawierającego rekord i długość rekordu. Funkcja może zapisać rekord krótszy od

podanej długości, gdy znak końca linii (\$9B) wystąpi w rekordzie na pozycji wcześniejszej niż zadeklarowana długość rekordu.

Oprócz adresu początkowego i długości rekordu funkcja PUT RECORD nie wymaga przekazania do wykorzystywanego bloku IOCB żadnych dodatkowych argumentów.

KOD FUNKCJI 11, PUT CHARACTERS
REALIZOWANE ZADANIE: Pisz znak lub znaki.

Funkcja PUT CHARACTERS zapisuje do pliku wyjściowego zawartość bufora pamięci, określonego przez adres początkowy i długość. Gdy podczas zapisu nie wystąpi błąd, to cała zawartość bufora jest przesyłana do pliku. Jeśli długość bufora pamięci jest równa 0, to funkcja PUT CHARACTERS przesyła jeden znak z akumulatora do pliku wyjściowego.

Oprócz adresu początkowego i długości bufora funkcja PUT CHARACTERS nie wymaga przekazania do wykorzystywanego bloku IOCB żadnych dodatkowych argumentów.

KOD FUNKCJI 12, CLOSE A FILE
REALIZOWANE ZADANIE: Zamknij plik.

Funkcja CLOSE A FILE zamyka plik oraz zwalnia wykorzystywany do jego obsługi blok IOCB. Jeśli plik otwarty był do zapisu, to przed jego zamknięciem zostanie do niego wprowadzona zawartość związanego z nim bufora pamięci.

KOD FUNKCJI 13. READ STATUS
REALIZOWANE ZADANIE: Czytaj status.

Do wywołania tej funkcji można użyć nieaktywnego bloku IOCB. Jej parametrem jest adres bufora zawierającego nazwę pliku. Jeśli plik o określonej nazwie nie istnieje lub jeśli jest nałożona na niego blokada zapisu, to zwracany jest odpowiedni kod błędu.

KOD FUNKCJI 32, RENAME A FILE
REALIZOWANE ZADANIE: Zmień nazwę pliku.

Parametrem tej funkcji jest adres bufora pamięci, zawierającego odpowiednio skonstruowany łańcuch znaków. Pierwsza część tego łańcucha jest nazwą pliku, która ma być zmieniona. Druga część, poprzedzona znakiem niedozwolonym w nazwie pliku, jest nową nazwą pliku. Pierwsza nazwa pliku może zawierać specyfikację napędu dyskowego i katalogu, w którym jest zapisany plik, którego nazwa ma być zmieniona. Nowa nazwa pliku nie może zawierać powyższych elementów. Do separacji nazw pliku: starej i nowej, najlepiej jest używać przecinka. Jest to znak niedozwolony w nazwie pliku. Na przykład: łańcuch „D2:TEST>PGMS>A.OUT,ZPCY” powoduje zmianę nazwy pliku A.OUT na ZPCY. Plik, którego nazwa ma być zmieniona jest zapisany w napędzie 2 w podkatalogu TEST>PGMS>. Należy zauważyć, że aby zmienić nazwę podkatalogu PGMS na MYBASIC, trzeba użyć łańcucha „D2:TEST>PGMS,MYBASIC”.

Wzorzec nazwy może być użyty jedynie w odniesieniu do nazwy pliku (tzn. za ostatnim znakiem „:” lub „>” występującym w łańcuchu). Efekt jego użycia najlepiej ilustruje następujący przykład: łańcuch znaków „D2:TEST:*.*,*.XYZ” powoduje zmianę nazwy wszystkich plików zapisanych w katalogu TEST napędu 2. Każda zmieniona nazwa pliku będzie miała rozszerzenie „.XYZ”. Jeśli w katalogu tym są zapisane pliki o nazwach: „ATEST.BAS”, „LOG” i „RAPORT.XYZ”, to po wykonaniu funkcji z podaniem parametru wskazującego na przedstawiony wyżej łańcuch, nazwy tych plików będą następujące: „ATEST.XYZ”, „LOG.XYZ” i „RAPORT.XYZ”.

KOD FUNKCJI 33, DELETE A FILE
REALIZOWANE ZADANIE: Skasuj plik.

Parametrem tej funkcji jest adres bufora pamięci, zawierającego nazwę pliku, który ma być skasowany. Plik, na który została nałożona blokada zapisu nie zostanie skasowany. Również katalog, który nie jest pusty nie zostanie

skasowany. Użycie funkcji do skasowania wymienionych wyżej plików lub podkatalogów generuje odpowiedni numer błędu.

Plik po skasowaniu może zostać odzyskany. Można w tym celu użyć jednego z wielu programów narzędziowych realizujących funkcje odzyskiwania skasowanych plików. Odzyskanie pliku będzie skuteczne jedynie w sytuacji, gdy po skasowaniu pliku zajmowane przez niego sektory nie zostały wypełnione nowymi informacjami.

KOD FUNKCJI 34, CREATE DIRECTORY

REALIZOWANE ZADANIE: Utwórz podkatalog.

Funkcja o kodzie 34 tworzy podkatalog o określonej nazwie na wskazanym napędzie dyskowym i we wskazanym podkatalogu. Gdy napęd dyskowy ani podkatalog nadrzędny nie zostaną określone, to funkcja utworzy podkatalog na roboczym napędzie i w roboczym podkatalogu.

Funkcji tej nie można użyć do utworzenia na dyskietce katalogu głównego (katalog główny dyskietki wskazywany jest przez podanie tylko oznaczenia napędu, np. „D1”). Parametrem tej funkcji jest przesłany do bloku IOCB adres bufora, zawierającego nazwę podkatalogu, który ma być utworzony. Bajty AUX1 i AUX2 powinny zawierać wartości wymagane do otwarcia pliku (patrz opis funkcji o kodzie 3); zwykle AUX1 = 8 i AUX2 = 0. Funkcja ta nie zostanie wykonana, gdy nazwa katalogu, który ma być utworzony jest identyczna z nazwą katalogu roboczego. Wywołanie funkcji CREATE DIRECTORY jest zwykle poprzedzone wywołaniem funkcji o kodzie 41, określającej katalog roboczy. Do utworzenia podkatalogu można również użyć funkcji o kodzie 42 wymagającej identycznych parametrów.

KOD FUNKCJI 35, LOCK FILE

REALIZOWANE ZADANIE: Nałóż blokadę zapisu na plik.

Funkcja o kodzie 35 nakłada blokadę zapisu na określony plik. Jej parametrem jest przesłany do IOCB adres bufora, zawierającego nazwę pliku, na który ma być nałożona blokada zapisu (lub wzorzec nazwy, gdy operacja taka ma być wykonana jednocześnie na wielu plikach).

Funkcja zostanie wykonana poprawnie również na pliku, na który wcześniej została nałożona blokada zapisu. Status pliku w takim przypadku nie zostanie zmieniony.

KOD FUNKCJI 36, UNLOCK FILE

REALIZOWANE ZADANIE: Zdejmij z pliku blokadę zapisu.

Funkcja UNLOCK FILE zdejmuje z pliku nałożoną blokadę zapisu. Wymaga ona parametrów identycznych jak funkcja LOCK FILE i jest tak samo realizowana.

KOD FUNKCJI 37, POINT TO POSITION IN FILE

REALIZOWANE ZADANIE: Ustaw wskaźnik pliku.

Wskaźnikiem pliku jest aktualny adres dyskowy bajtu z tego pliku, który zostałby odczytany lub zapisany z/do pliku, gdyby operacja taka została wykonana. Parametrem tej funkcji jest 3 bajtowy adres dyskowy, określający pozycję bajtu w pliku. Adres ten powinien być przesłany do bajtów 12 - 14 bloku IOCB. Funkcja może być wykonana tylko na plikach, które zostały otwarte do odczytu (tryb otwarcia 4) lub aktualizacji (tryb otwarcia 12). Pierwsze dwa bajty adresu dyskowego określają numer sektora (młodszy bajt i starszy bajt), a trzeci bajt określa numer bajtu w sektorze (offset). Po wykonaniu tej funkcji, odczytany bajt z pliku jest tym samym bajtem, który byłby odczytany lub zapisany do pliku po wykonaniu funkcji o kodzie 38, zwracającej identyczny adres dyskowy. Odpowiednikiem tej funkcji w języku BASIC jest instrukcja POINT. Podczas jej wykonywania na dysku o dużej pojemności (np. dysk twardy) może powstać problem. Wynika on z ograniczenia, które narzuca BASIC na numer sektora, będący jej parametrem (numer ten nie może być większy od 32767). W rezultacie instrukcja POINT nie jest przydatna do ustawienia wskaźnika pliku zapisanego w napędzie o pojemności przekraczającej 8MB. Problem ten można rozwiązać zastępując instrukcję POINT przedstawioną niżej sekwencją instrukcji:

```
OPEN #K,AUX1,AUX2,"D5:DUZYPLIK"  
.....  
NOTE #K,SEKTOR,BAJT  
.....  
POKE 844+16*K,ASC(CHR$(SEKTOR))  
POKE 845+16*K,INT(SEKTOR/256)  
POKE 846+16*K,BAJT  
XIO 37,#K,AUX1,AUX2,"D:"
```

W instrukcjach tych należy zwrócić uwagę, aby wartość bajtów AUX1 i AUX2 użytych w instrukcji XIO była taka sama, jak w instrukcji OPEN.

KOD FUNKCJI 38. NOTE POSITION IN FILE
REALIZOWANE ZADANIE: Odczytaj wskaźnik pliku.

Funkcja NOTE POSITION IN FILE odczytuje aktualny adres dyskowy bajtu z pliku. Odczytany adres przesyłany jest do bajtów 12 - 14 bloku IOCB i można go wykorzystać do zmiany wartości wskaźnika pliku. Funkcja ta może być wykonana na plikach otwartych do odczytu, zapisu i aktualizacji. Trzy bajty przesłane w wyniku jej wykonania do bloku IOCB mają następujące znaczenie:

bajt 12 - młodszy bajt numeru sektora;
bajt 13 - starszy bajt numeru sektora;
bajt 14 - numer bajtu w sektorze (offset).

KOD FUNKCJI 39, LOAD MEMORY
REALIZOWANE ZADANIE: Wczytaj program do pamięci.

Funkcja LOAD MEMORY wczytuje do pamięci komputera program zapisany w pliku binarnym o formacie zgodnym z DOS 2.0. Plik taki może być utworzony poleceniem K, wygenerowany przez asembler AMAC, MAC65 lub dowolny inny. Adresy obszaru pamięci, do którego zostanie wczytany program są wyspecyfikowane w zawierającym go pliku. Parametrami, które należy przekazać funkcji przez blok IOCB są: adres bufora pamięci zawierającego nazwę pliku binarnego i jedna z wartości: 4, 5, 6 lub 7. Wartości te są przekazywane przez bajt AUX1 bloku IOCB i mają następujące znaczenie:

AUX1=4 - uaktywnia adresy inicjacji i startu wczytywanego programu. Oznacza to, że po przeczytaniu programu automatycznie aktywowane zostaną wszystkie zawarte w nim procedury inicjujące i program zostanie uruchomiony.
AUX1=5 - procedury inicjujące nie zostaną aktywowane, natomiast program zostanie po załadowaniu automatycznie uruchomiony.
AUX1=6 - procedury inicjujące programu zostaną aktywowane, jednak program po załadowaniu nie zostanie uruchomiony.
AUX1=7 - tekst programu zostanie wczytany do pamięci, jednak ani procedury inicjujące nie zostaną aktywowane, ani program nie zostanie uruchomiony. Identyczne zadanie może być wykonane przez wywołanie funkcji o kodzie 40.

KOD FUNKCJI 41, SET DEFAULT DIRECTORY
REALIZOWANE ZADANIE: Określ katalog roboczy.

Parametrem funkcji jest adres bufora pamięci zawierającego nazwę podkatalogu, który ma być katalogiem roboczym. Najpierw sprawdzane jest, czy podana nazwa jest nazwą podkatalogu. Jeśli tak, to podkatalog ten staje się podkatalogiem roboczym, tj. będzie on automatycznie przeszukiwany przez MYDOS-a, gdy w podanej nazwie pliku, na którym ma być wykonana jakakolwiek operacja, nie zostanie określony numer napędu dyskowego.

KOD FUNKCJI 254, FORMAT A DISKETTE
REALIZOWANE ZADANIE: Formatuj (dyskietkę lub dysk twardy).

Parametrem tej funkcji jest adres bufora pamięci, w którym jest zapisany numer napędu dyskowego, zawierający dyskietkę do sformatowania. Jeśli bajty AUX1 i AUX2 bloku IOCB zawierają zera, to parametry użyte do formatowania są pobierane z tabeli konfiguracji systemu, zdefiniowanej za pomocą polecenia O. Gdy bit 7 bajtu AUX2 jest jedynką, to dyskietka fizycznie nie jest sformatowana. Jedynie kasowane są z niej wszystkie pliki i katalogi i zapisywane potrzebne informacje systemowe (przy założeniu, że dyskietka jest już sformatowana). Pozostałe bity bajtów AUX1 i AUX2 mogą zawierać 15-bitową liczbę sektorów na dyskietce.

Dyskietka może być sformatowana ze zwiększoną gęstością przez wpisanie do bajtu AUX1 wartości 1 (format taki nie jest kompatybilny z analogicznym formatem DOS-a 2.5).

VII. FORMATY DyskiETEK PRZETWARZANE PRZEZ MYDOS-a

Pierwsze trzy sektory dyskietki MYDOS wykorzystuje do zapisania informacji o jej formacie i krótkiego programu wczytującego DOS-a do pamięci komputera. Program taki jest zapisywany tylko na tych dyskietkach, które zawierają pliki DOS.SYS i DUP.SYS. Sektor o numerze \$168 (oraz sektory o numerach \$167, \$166, \$165, itd. - dysków o dużej pojemności) jest wykorzystywany na bitową mapę wolnych sektorów na dyskietce. Sektor ten (lub sektory) zawiera tabelę zawartości wolumenu, zwaną w skrócie VTOC. Kilka bajtów tabeli VTOC zawiera informacje o formatach plików zapisanych na dyskietce. Sektory \$169 i \$170 zawierają katalog główny dyskietki. W katalogu tym zapisane są nazwy plików i podkatalogów, ilość sektorów zajmowanych przez pliki i podkatalogi oraz początkowe numery sektorów plików i podkatalogów.

Z przedstawionego opisu wynika, że gdy dyskietka jest jednostronna i o pojedynczej gęstości (720 sektorów), to format tworzony przez MYDOS-a jest prawie identyczny z formatem tworzonym przez DOS 2.0. Jedyna różnica polega na tym, że DOS 2.0 nie wykorzystuje sektora o numerze 720, natomiast MYDOS używa go. W rezultacie, gdy dyskietka zostanie sformatowana, MYDOS informuje, że jest na niej 708 wolnych sektorów, a DOS 2.0, że takich sektorów jest 707.

Zasadnicze różnice pojawiają się, gdy za pomocą MYDOS-a zostanie sformatowany dysk o dużej pojemności. W takim przypadku wzrasta liczba sektorów o numerach mniejszych od \$168 zarezerwowanych na tabelę VTOC. Katalog główny takiego dysku może być odczytany przez DOS 2.0, jednak informacje zapisane w plikach utworzonych na dysku o dużej pojemności mogą być odczytane jedynie przez MYDOS-a. DOS 2.0 mógłby z takiego dysku odczytać tylko te pliki, które zajmują pierwsze 1023 sektory i tylko wtedy, gdyby wyeliminowana została realizowana przez niego kontrola numeru pliku.

Zgodność formatu tworzonych przez MYDOS-a z formatem używanym przez DOS 2.0 maleje jeszcze bardziej, gdy na dyskietce tworzone są podkatalogi. DOS 2.0 traktuje podkatalogi jako proste pliki o nie dającej się odczytać zawartości. Z tego powodu dyskietki z oprogramowaniem, sprzedawane przeciętnemu nabywcy, nie powinny zawierać podkatalogów, chyba że zostanie wyraźnie zaznaczone, że mogą być one wykorzystane jedynie pod kontrolą MYDOS-a. Problem wymiany informacji na dyskietkach komplikuje się jeszcze bardziej po uwzględnieniu istnienia wielu formatów stosowanych przez różnych producentów napędów dyskowych do komputerów Atari. W rezultacie do odczytania informacji zapisanych na dyskietce dwustronnej z podwójną gęstością nie wystarcza komputer i system MYDOS. Trzeba dodatkowo wiedzieć, jaki system został użyty do zapisania danych na dyskietkę, czy PERCOM, SWP, Astra, Supra, czy inny.

VIII. WYKORZYSTANIE PAMIĘCI PRZEZ MYDOS-a

Rezydentna część MYDOS-a zajmuje obszar pamięci od adresu \$0700 do \$1EE9. Dodatkowo, jeśli aktywne jest menu MYDOS-a, to zajęta jest pamięć od adresu \$294A do \$4371. W tym przypadku MYDOS wykorzystuje również pierwszych szesnaście bajtów obszaru pamięci, które zajmowane są przez procedury realizujące arytmetykę zmiennoprzecinkową (bajty o adresach \$D4 - \$E3). W odróżnieniu od DOS 2.0, program obsługi plików MYDOS-a wykorzystuje niektóre procedury arytmetyki zmiennoprzecinkowej z OS ROM. Nierezydentna część MYDOS-a (program zapisany w

pliku DUP.SYS) wczytywany jest do pamięci poczynając od adresu \$294A. Program ten rezerwuje obszar pomiędzy adresami \$1EE9 - \$2949 na buforów związane z przetwarzanymi plikami i programowe sterowniki urządzeń. Przy zarezerwowaniu trzech buforów przeznaczonych do obsługi plików, na sterowniki programowe urządzeń pozostaje około 2500 bajtów.

IX. PRZYSTOSOWANIE SYSTEMU DO WYMAGAŃ UŻYTKOWNIKA

KONFIGURACJA RAM-DYSKU

Program obsługi RAM-dysku, dostępny w MYDOS-ie wersja 4.50, automatycznie instaluje RAM-dysk w rozszerzeniach pamięci kompatybilnych ze standardem zastosowanym w komputerze Atari 130XE. Istnieje jednak możliwość takiego zdefiniowania konfiguracji systemu, aby odpowiadała ona w maksymalnym stopniu potrzebom użytkownika. Jest to szczególnie istotne, gdy rozszerzenie pamięci nie jest podzielone na 64kB banki, jak ma to miejsce w Atari 130XE. Na przykład rozszerzenie AXLON wymaga zupełnie innego sposobu zarządzania pamięcią.

Plik RAMBOOT.M65 zapisany na dyskietce systemowej MYDOS-a zawiera kod źródłowy dla asemblera MAC/65. Kod wynikowy uzyskany po asemblacji zapisany jest w pliku RAMBOOT.AUT. Jeśli nazwa tego pliku zostanie zmieniona na AUTORUN.SYS, to MYDOS wykorzystywał będzie RAM-dysk w sposób identyczny jak DOS 2.5 używał RAM-dysku na komputerze Atari 130XE, tzn. do RAM-dysku zostanie skopiowany plik DUP.SYS, RAM-dysk zostanie skojarzony z napędem o numerze 8 i z napędu o tym numerze wczytywany będzie do pamięci zarówno plik DUP.SYS jak i MEM.SAV. Można wygenerować nowy plik AUTORUN.SYS przez modyfikację kodu źródłowego zapisanego w pliku RAMBOOT.M65 i jego asemblację. Dzięki temu istnieje możliwość kojarzenia RAM-dysku z napędem o numerze innym niż 8 oraz kopiowania do niego wszystkich plików, które użytkownik zechce w nim zapisać. RAM-dysk instalowany automatycznie przez MYDOS-a wykorzystuje 64kB bank pamięci i udostępnia użytkownikowi 499 sektorów o długości 128 bajtów każdy. Polecenie „O” stanowi prosty sposób modyfikacji programu obsługi RAM-dysku, który jest instalowany w rozszerzeniu pamięci podzielonym na 64kB banki. Większość rozszerzeń pamięci produkowanych dla komputerów Atari 800XL i 130XE wykorzystuje ten sam rejestr do sterowania dostępem do pamięci. Jest nim PORTB układu PIA. RAM-dysk o pojemności 128kB może być zainstalowany na komputerze Atari 130XE przez wykorzystanie ostatniego z nie używanych bitów tego portu. Wybór 64kB banku jest realizowany za pomocą bitu 6 portu PORTB. Zdefiniowana w MYDOS-ie tabela sekwencji stron pamięci jest wybierana przez przypisanie wartości 2 odpowiedniemu parametrowi instalacyjnemu (patrz opis polecenia O). Ten sam efekt uzyskuje się przez naciśnięcie klawisza „Y” na wyświetlone pytanie: „Use default page sequence...?”

Jeśli rozszerzenie pamięci nie jest wykonane przez dodanie jednego lub dwóch 64kB banków, lecz stanowi pojedynczy blok pamięci o pojemności 256kB, to obszar 192kB musi być podzielony na banki. W takim przypadku do wskazania banku pamięci potrzebne są 4 bity portu PORTB. Często w tym celu wykorzystywane są bity 0 i 1 (w Atari 130XE), bit 6 (w rozszerzeniu opisanym wyżej) i bit 5 (w Atari 130XE jest on wykorzystywany do przełączania banków pamięci zawierających dane obrazu). Programy wykorzystujące jednocześnie dodatkową pamięć do własnych celów i do przechowywania danych tworzących obraz nie będą działać z rozszerzeniami tego typu. Przyczyną jest brak skutecznego mechanizmu zezwalającego na wielofunkcyjne wykorzystanie bitów rejestru PORTB (ten sam bit musiałby wskazywać bank używany przez procesor centralny komputera i bank wykorzystany przez procesor graficzny ANTIC). Przedstawione wyżej cechy ma rozszerzenie NEWELL INDUSTRIES 256K, produkowane dla komputerów Atari 800XL.

Jeśli dodatkowa pamięć jest dołączona do komputera przez wyprowadzone gniazdo rozszerzeń, to sterowanie dostępem do niej wymaga istnienia nowego rejestru. W ten sposób pracuje rozszerzenie Axlon RAMPOWER 128 dla komputerów Atari 800. W rozszerzeniach tego typu wybór 16kB strony pamięci realizowany jest przez wpisanie numeru strony do rejestru sterującego dostępem do rozszerzenia. Aby MYDOS mógł współpracować z rozszerzeniami tego typu, użytkownik musi określić adres rejestru sterującego i tabelę sekwencji stron rozszerzenia (dla RAMPOWER 128 właściwa tabela jest identyfikowana przez wartość 5).

W MYDOS-ie są zakodowane 4 tabele sekwencji stron. Każda z nich może zawierać nie więcej niż 32 numery stron pamięci rozszerzenia. Podczas konfiguracji RAM-dysku użytkownik może wybrać tabelę sekwencji stron, która będzie wykorzystywana przez MYDOS-a, poprzez przypisanie wartości 0, 1, 2 lub 5 odpowiedniemu parametrowi konfiguracyjnemu. Jeśli wykorzystywane rozszerzenie jest zgodne z rozszerzeniem zastosowanym w komputerze Atari 130XE, to proces konfigurowania RAM-dysku może być pominięty. W takiej sytuacji MYDOS użyje tzw. domyślnych parametrów konfiguracyjnych, które są w nim zapisane. Jeśli użytkownik wprowadza własną sekwencję stron, to musi pamiętać, że liczba wprowadzonych numerów stron określa jednocześnie rozmiar pamięci użytej na RAM-dysk.

Tabele sekwencji stron „zaszyte” w MYDOS-ie i identyfikujące je wartości są następujące:

Wartość identyfikująca tabelę	Sekwencja numerów stron	Wartość używana do odtworzenia zawartości rejestru sterującego dostępem
0	E3,E7,EB,EF,C3,C7,CB,CF,83,87,8B,8F,A3,A7,AB,AF	00
1	C3,C7,CB,CF,83,87,8B,8F,E3,E7,EB,EF,A3,A7,AB,AF	00
2	A3,A7,AB,AF,C3,C7,CB,CF,E3,E7,EB,EF,83,87,8B,8F	00
5	00,01,02,03,04,05,06,07,08,09,0A,0B,0C,0D,0E,0F	FF

Na przykład: jeśli używane jest rozszerzenie pamięci RAMBO XL i strony o numerach E3, E7, EB i EF mają być zarezerwowane dla BASIC-a XE (w trybie extended), to definiując konfigurację RAM-dysku należy określić jego rozmiar na 128kB i przypisać 1 wartości identyfikującej tabelę lub wprowadzić następującą sekwencję stron: C3,C7,CB,CF,83,87,8B,8F,00.

LICZBA JEDNOCZEŚNIE OTWARTYCH PLIKÓW

Liczba plików, które mogą być jednocześnie otwarte jest zapisana, podobnie jak w DOS-ie 2.0, w bajcie o adresie \$0709 (dziesiętnie 1801). Bajt ten może zawierać wartość od 1 do 16. Standardową zapisaną wartością jest 3, to najmniejsza liczba jednocześnie otwartych plików, umożliwiającą wykonanie wszystkich poleceń MYDOS-a (kopiowanie pliku wymaga jednoczesnego otwarcia trzech plików). Aby zmienić wartość tego bajtu, należy użyć polecenia „O”. Aby wprowadzoną zmianę zapisać na stałe do MYDOS-a, trzeba po jej wykonaniu zapisać system na dysk (polecenie „H”). Należy pamiętać, że dla każdego pliku rezerwowany jest przez MYDOS-a 256 bajtowy bufor.

DEFINIOWANIE NAPĘDÓW DYSKOWYCH WSPÓŁPRACUJĄCYCH Z MYDOS-em

Podobnie jak DOS 2, MYDOS automatycznie identyfikuje liczbę stacji dysków podłączonych do komputera podczas włączania lub restartu systemu. Standardowo w MYDOS-ie zdefiniowane są dwa napędy. Jeśli użytkownik chce zdefiniować więcej napędów, może w tym celu użyć polecenia „O”. Jeżeli zdefiniowane napędy mają być na stałe wpisane do MYDOS-a, należy po ich zdefiniowaniu zapisać system na dysk poleceniem „H”.

REALIZACJA ZAPISU NA DYSK Z WERYFIKACJĄ LUB BEZ

Standardowo MYDOS realizuje zapis na dysk bez weryfikacji. Dzięki temu zapis odbywa się znacznie szybciej. Przez użycie polecenia „O” można „nakazać” MYDOS-owi weryfikowanie każdej operacji zapisu na dysk. Należy pamiętać, że zapis z weryfikacją trwa około trzy razy wolniej niż zapis bez weryfikacji.

O tym, czy MYDOS ma realizować operacje zapisu z weryfikacją, czy bez, decyduje wartość wpisana do bajtu o adresie \$0779 (dziesiętnie 1913). Wykorzystując ten bajt, można z poziomu języka BASIC sterować sposobem zapisu. Na przykład: instrukcja POKE 1913,87 powoduje, że zapis będzie wykonywany z weryfikacją, a instrukcja POKE 1913,80 nakazuje realizowanie zapisu bez weryfikacji.

X. WSPÓŁPRACA Z NAPĘDEM DYSKOWYM

Napędy dysków elastycznych dołączane są do 8-bitowych komputerów Atari przez łącze szeregowo, którego gniazdo umieszczone jest w obudowie komputera (na prawej stronie lub na tylnej ścianie). Stacje dysków o dużej pojemności (dysk twardy) można dołączyć przez port łącza równoległego, w który wyposażone są i komputery Atari 800XL i 130 XE. Do obsługi urządzeń współpracujących z komputerem przez łącze szeregowo jest wykorzystywana procedura systemu operacyjnego nazwana akronimem SIO (Serial Input/Output routine). Procedura ta jest używana również przez MYDOS-a. Do wskazania procedurze SIO jaką operację ma wykonać (komendy do wykonania) i na jakim urządzeniu, służy blok sterowania urządzeniem DCB (Device Control Block). Jest to obszar pamięci o adresie początkowym \$0300. Firma Atari zaimplementowała w procedurze SIO podstawowy zestaw komend, niezbędnych do obsługi napędu Atari 810. Jednak większość producentów bardziej funkcjonalnych napędów dyskowych zestaw ten znacznie rozszerzyło. MYDOS może współpracować z każdą stacją dysków, która jest kompatybilna z napędem 810. Współpraca z napędami o lepszych parametrach wymaga istnienia dodatkowych komend np. komendy umożliwiającej identyfikację gęstości zapisu dyskietki podczas odczytu sektora o numerze 1. Jest to szczególnie konieczne, gdy w tej samej stacji dysków będą odczytywane dyskietki sformatowane z pojedynczą lub podwójną gęstością zapisu.

Minimalny zestaw komend, które powinny być wykonywalne na napędzie dyskowym, aby mógł on współpracować z MYDOS-em, jest następujący:

Kod urządzenia	Urządzenie	Kod komendy	Kierunek transmisji	Liczba bajtów	Bajty pomocnicze	Operacja
\$31	D#	\$21	z napędu	128/256	0	formatowanie
\$31	D#	\$50	do napędu	128/256	1-720	zapis bez weryfikacji
\$31	D#	\$52	z napędu	128/256	1-720	odczyt
\$31	D#	\$53	z napędu	4	0	odczyt statusu
\$31	D#	\$57	do napędu	128/256	1-720	zapis z weryfikacją

Ponadto, aby uzyskać format dyskietki o zwiększonej gęstości zapisu musi być wykonywalna komenda:

Kod urządzenia	Urządzenie	Kod komendy	Kierunek transmisji	Liczba bajtów	Bajty pomocnicze	Operacja
\$31	D#	\$22	z napędu	128	0	formatowanie

Liczba bajtów jest zawsze równa 128 dla napędów umożliwiających zapis w pojedynczej gęstości i ma tę samą wartość dla sektorów 1, 2 i 3 w napędach o podwójnej gęstości zapisu. Wszystkie pozostałe sektory w tych napędach mają długość 256 bajtów.

Komenda realizująca formatowanie dyskietki jest zawsze wywoływana z numerem sektora z przedziału 4 - 720. Długość sektora (liczba bajtów) należy ustawić na 128, gdy uzyskana ma być pojedyncza gęstość zapisu, a na 256, gdy formatowanie ma być wykonane z podwójną gęstością.

Pierwszy bajt zwracany przez komendę „ODCZYT STATUSU” wskazuje rozmiar sektora. Jeśli ustawiony jest jego piąty bit, to sektor jest 256 bajtowy, w przeciwnym wypadku sektor jest 128 bajtowy.

Bajty pomocnicze zawierają adres (numer) sektora. Jest nim liczba z przedziału 1 - 720 lub 1 - 65535. Drugi przypadek dotyczy jedynie dysków o bardzo dużej pojemności.

Dodatkowo, jeśli stacja dysków może mieć dynamicznie zmieniane parametry, to jej sterownik musi być w stanie wykonać następujące komendy:

Kod urządzenia	Urządzenie	Kod komendy	Kierunek transmisji	Liczba bajtów	Bajty pomocnicze	Operacja
\$31	D#	\$4E	z napędu	12	1-720	Odczyt konfiguracji
\$31	D#	\$4F	do napędu	12	1-720	Zapis konfiguracji

Jeśli sterownik i stacja dysków nie jest w stanie wykonać powyższych komend, to stacja taka powinna być zdefiniowana w systemie jako napęd, którego konfiguracja jest niemożliwa. Przedstawione komendy wykorzystywane są przez polecenie „P” MYDOS-a.

Znaczenie 12 bajtów przesyłanych z i do stacji przez opisane wyżej dwie komendy jest następujące:

Numer bajtu	Znaczenie
0	Liczba ścieżek na stronie (40 dla standardowego napędu).
1	Czas przesuwu głowic między sąsiednimi ścieżkami. Czas ten należy określać zgodnie z definicją firmy Western Digital.
2	Liczba sektorów na ścieżce (starszy bajt). Zwykle 0.
3	Liczba sektorów na ścieżce (młodszy bajt). Dla standardowej dyskietki - 18.
4	Kod liczby stron. 0 - dyskietka jednostronna, 1 - dwustronna.
5	Kod typu napędu: bit 2 = 0 - napęd o pojedynczej gęstości zapisu; bit 2 = 1 - napęd o podwójnej gęstości zapisu; bit 1 = 0 - napęd 5,25 calowy; bit 1 = 1 - napęd 8 calowy.
6	Liczba bajtów w sektorze (starszy bajt). Dla pojedynczej gęstości - 0.
7	Liczba bajtów w sektorze (młodszy bajt). Dla pojedynczej gęstości - 128.
8	Bajt sterowania transmisją: bit 7 = 1 - dyskietka 40 ścieżkowa w napędzie 80 ścieżkowym; bit 6 = 1 - jest zawsze ustawiony, gdy napęd jest podłączony; bit 1 = 1 - nakazuje traktować sektory 1, 2 i 3 jako pełne; bit 0 = 1 - wskazuje, że sektory numerowane są od 0, a nie od 1.
9-11	Bajty te nie są wykorzystywane przez MYDOS-a. Należy jednak sprawdzić w dokumentacji używanego napędu, jakie wartości powinny one zawierać.

MYDOS zawsze odczytuje konfigurację napędu przed zapisaniem nowych parametrów. Zawartość bajtów 9-11 jest powielana podczas tej operacji. w rezultacie zostaną one zapisane w niezmienionej postaci.

Użycie opisanych wyżej komend do dysku twardego wymaga dodatkowych zmian. Parametry konfiguracyjne dysku twardego są bardzo złożone i są zwykle na stałe wpisane w jego sterowniku lub zapisane w jakimś „tajemniczym” jego obszarze. Aby podzielić dysk twarde na partycje (dotyczy to dysków o pojemności przekraczającej 16MB), MYDOS używa komendy zmieniającej konfigurację z

parametrem: „liczba sektorów na ścieżce” równym liczbie sektorów na dysku (patrz opis polecenia O) i liczbę ścieżek ustawia na 1.

XI. WSPÓŁPRACA Z RAM-DYSKIEM

Istniejący w MYDOS-ie programowy sterownik napędów dyskowych został tak opracowany, że wyeliminowano potrzebę utworzenia dodatkowego sterownika do obsługi RAM-dysku. Aby odczytać lub zapisać sektor danych w rozszerzeniu pamięci RAM, sterownik ten wykonuje następujące operacje:

1. Numer sektora jest dzielony przez 128. Reszta z dzielenia zostaje pomnożona przez 128 i do iloczynu dodawana jest liczba 16384. Otrzymana wartość jest adresem pierwszego bajtu sektora w obszarze pamięci adresowanym wartościami \$4000 - \$7FFF (dziesiętnie 16384 - 32767).

2. Wyznaczony wyżej iloraz jest wykorzystywany jako indeks wskazujący pozycję w tabeli sekwencji stron pamięci. Każda pozycja tej tabeli określa adres 16kB obszaru pamięci, który można odwzorować w lokacjach o adresach \$4000 - \$7FFF.

3. Na wartościach pobranych z tabeli sekwencji stron i z rejestru (bajtu) sterującego dostępem do rozszerzenia pamięci wykonywana jest operacja koniunkcji (AND). Otrzymana w jej wyniku wartość jest zapisywana w rejestrze sterującym dostępem do rozszerzenia pamięci.

4. Obliczony w powyższy sposób adres wskazuje miejsce w rozszerzeniu pamięci, zarezerwowane na sektor o określonym numerze. Dane zapisane w sektorze są podczas odczytu przenoszone do bufora lub gdy sektor ma być zapisany, dane przenoszone są z bufora pod wyznaczony adres.

5. Odtwarzana jest zawartość rejestru sterującego dostępem do rozszerzenia pamięci. Odtworzenie polega na wykonaniu operacji OR na zawartości rejestru i ostatniej wartości z tabeli sekwencji stron. Uzyskana w jej wyniku wartość wpisywana jest do rejestru.

Wynika stąd, że gęstość zapisu stosowana w RAM-dysku jest pojedyncza (długość sektora wynosi 128 bajtów), a maksymalna jego pojemność wynosi 4MB (256 stron po 16384 bajtów każda). Jednak MYDOS jest w stanie obsłużyć RAM-dysk o pojemności nie przekraczającej 1MB, gdyż zarezerwowana w nim tabela sekwencji stron ma długość tylko 64 bajtów.

Jak wynika z przedstawionej procedury wyznaczania adresu sektora w RAM-dysku, do jej wykonania potrzebne są następujące parametry:

- adres rejestru sterującego dostępem do rozszerzenia pamięci. Dla komputerów Atari 130XE jest to adres \$D301, dla rozszerzeń typu Axlon adres ten jest równy \$CFFF;
- wartość używana do odtworzenia zawartości rejestru sterującego dostępem do rozszerzenia pamięci. Wartość ta jest równa \$00 dla rozszerzenia użytego w Atari 130XE, a \$FF dla rozszerzenia typu Axlon;
- tablica sekwencji stron pamięci rozszerzenia;
- numer sektora.

Tablica sekwencji stron jest tworzona w następujący sposób: najpierw określone są bity rejestru sterującego dostępem do rozszerzenia pamięci, które nie mogą być zmieniane. W każdym bajcie tabeli sekwencji stron pamięci bity o określonych wyżej numerach ustawiane są na 1. Pozostałe bity 2-3 i 5-6 muszą zawierać numery 12 stron rozszerzenia pamięci. Sposób ich wypełnienia przedstawia przykład. Ilustruje on sposób tworzenia wartości tabeli sekwencji stron dla rozszerzenia NEWELL INDUSTRIES 256K. Sposób zarządzania pamięcią tego rozszerzenia odpowiada standardowi użytemu w Atari 130XE. Bitami, które nie mogą być zmieniane w rejestrze sterującym dostępem do tego rozszerzenia są bity o numerach 7, 1 i 0.

Bity: **7 6 5 4 3 2 1 0**
1 x x x x x 1 1

Taka wartość jest wpisywana do każdego bajtu tabeli sekwencji stron. Następnie bity oznaczone „x” wypełniane są kombinacją zer i jedynek. W ten sposób

tworzonych jest 12 wartości tabeli sekwencji stron. Oto one:

Bity:	7	6	5	4	3	2	1	0
	1	0	0	0	0	0	1	1
	1	0	0	0	0	1	1	1
	1	0	0	0	1	0	1	1
	1	0	0	0	1	1	1	1
	1	1	0	0	0	0	1	1
	1	1	0	0	0	1	1	1
	1	1	0	0	1	0	1	1
	1	1	0	0	1	1	1	1
	1	1	1	0	0	0	1	1
	1	1	1	0	0	1	1	1
	1	1	1	0	1	0	1	1
	1	1	1	0	1	1	1	1

Ostatnim, 13 elementem tabeli sekwencji stron dla rozszerzenia NEWELL INDUSTRIES 256K jest zero. Zatem sekwencja ta ma postać: 83, 87, 8B, 8F, C3, C7, CB, CF, E3, E7, EB, EF, 0.

XII. INSTRUKCJA INSTALOWANIA MYDOS-a

Poniżej przedstawiono dokładny opis czynności, które należy wykonać w celu zainstalowania MYDOS-a na komputerze. Do wykonania instalacji potrzebna jest dyskietka z oryginalnym systemem MYDOS (nie może to być system zmodyfikowany). Czynności związane z instalowaniem powinny być wykonane w takiej kolejności, w jakiej zostały opisane.

1. Przed przystąpieniem do instalowania, należy sprawdzić, czy do komputera nie jest dołączony żaden zewnętrzny moduł pamięci ROM (kartridż).
2. Włącz stację dysków i włóż do niej dyskietkę zawierającą pliki systemowe MYDOS-a.
3. Włącz komputer (na komputerach XL i XE należy podczas włączania przytrzymać wciśnięty klawisz OPTION). Po kilku sekundach zostanie wyświetlone menu MYDOS-a i związany z nim komunikat zachęty: „Select Item (RETURN for menu)”.
4. Jeśli posiadasz komputer z dwiema stacjami dysków, z których każda jest jednostronnym napędem umożliwiającym zapis 40 ścieżkowy i nie masz rozszerzenia pamięci (ani Twój komputer nie jest komputerem Atari 130XE), kontynuuj instalowanie systemu od punktu 6.
5. Określ konfigurację każdego z posiadanych napędów dyskowych tylko wtedy, gdy są nimi: dysk twardy, napęd XF-551 lub dowolny inny umożliwiający zapisanie na stronie dyskietki więcej niż 40 ścieżek. Użyj do tego polecenia „O” z menu MYDOS-a.
6. Jeśli chcesz, aby w Twoim systemie można było otworzyć jednocześnie więcej niż trzy pliki lub chcesz zmienić którykolwiek z jego parametrów konfiguracyjnych, wykonaj polecenie „O”. Jeśli wykorzystujesz w systemie RAM-dysk o standardzie innym niż użyty w komputerze Atari 130XE, to również musisz go zdefiniować za pomocą polecenia „O”****.
7. Następnie włóż do stacji dysków czystą dyskietkę i sformatuj ją poleceniem „I”. Jeśli podczas formatowania zostanie zasygnalizowany błąd, użyj nowej dyskietki, gdyż pierwsza najprawdopodobniej jest uszkodzona.
8. Zapisz na sformatowaną dyskietkę pliki systemowe DOS.SYS i DUP.SYS poleceniem „H”. Następnie, jeśli nie wykorzystujesz RAM-dysku i pracujesz na komputerze innym niż Atari 130XE, kontynuuj instalowanie systemu od punktu 10.
9. Jeśli pracujesz na komputerze Atari 130XE lub masz jakiegokolwiek inne rozszerzenie pamięci, w którym zechcesz zainstalować RAM-dysk, skopiuj plik RAMBOOT3.AUT z dyskietki systemowej na dyskietkę, na którą skopiowałeś pliki systemowe MYDOS-a. Zmień nazwę skopiowanego pliku na AUTORUN.SYS. W tym celu wykonaj polecenie „C” z parametrem "RAMBOOT3.AUT,AUTORUN.SYS/X**.

10. W ten sposób zainstalowałeś MYDOS-a na Twoim komputerze. Zabezpiecz przed zapisem dyskietkę, na której zainstalowałeś system. Będzie to robocza kopia systemu, której powinieneś używać zawsze przy włączaniu komputera. Za pomocą polecenia „J” możesz wykonać dowolną liczbę kopii tej dyskietki.

UWAGI.

Program RAMBOOT3.AUT formatuje najpierw RAM-dysk a następnie kopiuje do niego zawartość podkatalogu RAMDISK, gdy istnieje taki na dyskietce, z której ładowany jest system. Za pomocą polecenia „Q” możesz utworzyć na dyskietce podkatalog o takiej nazwie i skopiować do niego poleceniem „C” wszystkie pliki, które chcesz, aby były w RAM-dysku. Jeśli nie chcesz, aby ta procedura była wykonywana przy każdorazowym włączeniu komputera, nie wykonuj czynności opisanej w punkcie 9.

** — Na dyskietce systemowej znajduje się plik zawierający kod źródłowy (dla assemblera) programu kopiującego do RAM-dysku określone pliki. Jeśli użytkownik ma taką potrzebę, to może zmodyfikować ten kod w taki sposób, aby program kopiował oprócz pliku DUP.SYS również inne pliki lub automatycznie kojarzył RAM-dysk z napędem D8.

**** — Jeśli RAM-dysk jest instalowany w rozszerzeniu pamięci NEWELL INDUSTRIES dla Atari 800XL, to używana sekwencja stron może być ustalona przez MYDOS-a (tzw. domyślna sekwencja) lub określona przez wartość 0 lub 1. Strony E3/E7/EB/EF mogą być zarezerwowane dla BASIC-a XE (w trybie extended) przez ustalenie rozmiaru RAM-dysku na 128k i określenie sekwencji stron przez wartość 1. Adres rejestru sterującego dostępem do rozszerzenia pamięci jest równy \$D301. W takim przypadku rozszerzenie pamięci będzie bezkolizyjnie wykorzystywane zarówno przez RAM-dysk jak i BASIC XE.

MYDOS zakłada, że RAM-dysk będzie tworzony w rozszerzeniu pamięci spełniającym standard zastosowany w komputerach Atari 130XE. Jeśli RAM-dysk będzie instalowany w rozszerzeniu AXLON RAMPOWER 128 dla komputerów serii Atari 800, to należy wykonać następujące czynności:

1. Zablokować tryb pracy rozszerzenia pamięci, w którym dostępne jest 128kB.
2. Zainstalować MYDOS-a określając parametry konfiguracyjne RAM-dysku: rozmiar - 112kB, sekwencja stron - wartość 5, adres rejestru sterującego dostępem do rozszerzenia - \$CFFF.
3. Skopiować pliki systemowe MYDOS-a na nową, sformatowaną dyskietkę. Skopiować na nią plik AUTORUN.SYS.
4. Odblokować tryb pracy rozszerzenia pamięci, w którym dostępne jest 128kB.
5. Wykonać restart systemu z nowej dyskietki.

XIII. KODY BŁĘDÓW

Niżej przedstawione zostały kody błędów zwracane przez system operacyjny (OS ROM) i MYDOS-a. Gwiazdką oznaczono kody błędów niewystępujących w Atari DOS 2.0 i 2.5 lub mających inne znaczenie w MYDOS-ie. Większość z nich ma swoje odpowiedniki w starszych wersjach dyskowych systemów operacyjnych. Nowe kody błędów są rezultatem rozszerzonych możliwości MYDOS-a. Błąd o kodzie 164 dotyczy tylko plików mających format DOS 2.0 lub DOS 2.5. Błędy 180 i 181 mogą pojawić się tylko podczas wywołania procedury CIO z kodem funkcji 39 lub 40 (ładowanie programu do pamięci). Błędy 171 i 175 związane są z tworzeniem i kasowaniem podkatalogów i nie mają odpowiedników w DOS 2.0 i DOS 2.5; błąd 174 związany jest z dostępem do pliku, który zapisany jest w określonym katalogu i również nie ma odpowiednika w starszych wersjach dyskowych systemów operacyjnych. Błąd 173 ma identyczne znaczenie jak w DOS 2.0, lecz podczas pracy z MYDOS-em może występować znacznie częściej, wskazując dokładniej uszkodzone dyskietki.

Kod błędu	Znaczenie
3	Podczas realizacji odczytu osiągnięto koniec pliku (MYDOS).
128	Operacja przerwania przez naciśnięcie klawisza BREAK (OS ROM).
129	Próba otwarcia już otwartego bloku IOCB (OS ROM).
130	Użyte urządzenie nie jest zdefiniowane w systemie (OS ROM).
131	Próba odczytu danych, gdy IOCB został otwarty tylko do zapisu (OS ROM).
132	Niedozwolony kod funkcji (OS ROM).
133	Urządzenie lub plik nie zostało otwarte do operacji WE/WY.
134	Błędne użycie bloku IOCB (OS ROM).
135	Próba zapisu danych, gdy IOCB został otwarty tylko do odczytu (OS ROM).
136	Próba odczytu danych z pliku po wykryciu jego końca (MYDOS).
137	Niepełny rekord (OS ROM).
138	Urządzenie nie odpowiada (OS ROM).
139	Urządzenie odpowiada sygnałem NAK (potwierdzenie niezgodności). Błąd transmisji przez łącze szeregowe (OS ROM).
141	Współrzędne położenia kursora przekraczają wartości dopuszczalne dla aktualnie wybranego trybu graficznego (OS ROM).
142	Błąd transmisji przez łącze szeregowe (OS ROM).
143	Błędna suma kontrolna podczas transmisji przez łącze szeregowe (OS ROM).
144	Błąd podczas operacji WE/WY na urządzenie zewnętrzne (OS ROM).
146	Funkcja nie może być wykonana przez program obsługi urządzenia (OS ROM).
147	Zbyt mały obszar wolnej pamięci RAM (OS ROM).
160	Zły numer urządzenia np. stacji dysków (MYDOS i OS ROM).
161	Za dużo otwartych plików (MYDOS).
162	Dysk pełny. Brak miejsca dla pliku wyjściowego (MYDOS).
163*	Dysk zabezpieczony przed zapisem lub uszkodzony (MYDOS).
164	Bajt łącznikowy w sektorze wskazuje na sektor leżący w innym katalogu (MYDOS).
165	Niepoprawna nazwa pliku (MYDOS).
166	Instrukcja POINT wskazuje bajt nie leżący wewnątrz pliku (MYDOS).
167	Na pliku nie można wykonać operacji, gdyż jest nałożona na niego blokada zapisu (MYDOS).
168	Błędny numer bloku IOCB (MYDOS i OS ROM).
169	Zapełniony katalog lub podkatalog (MYDOS).
170	Wskazany plik nie znajduje się w katalogu (MYDOS).
171	Urządzenie lub plik nie zostało otwarte (MYDOS i OS ROM).
172*	Plik lub podkatalog o podanej nazwie już istnieje na dysku (MYDOS).
173	Dyskietyki nie można sformatować. Uszkodzona dyskietka lub napęd (MYDOS).
174*	Na dysku brak wskazanego podkatalogu (MYDOS).
175*	Katalog lub podkatalog nie jest pusty i nie może być skasowany (MYDOS).
180*	Plik nie zawiera programu w kodzie maszynowym (MYDOS).
181*	Źle określono adres początkowy i końcowy obszaru do którego program ma być wczytany - adres początkowy jest większy od adresu końcowego (MYDOS).

XIV. VTOC Fixer - PROGRAM KONTROLI TABELI ZAWARTOŚCI WOLUMENU

Czasami zdarza się, że nagle, bez określonej przyczyny, pliku zapisanego na dysku twardym nie da się odczytać lub podczas próby dostępu do niego jest sygnalizowany błąd. Dotyczyć to może również plików zapisanych na dyskietce. W takiej sytuacji pomocnym może okazać się program kontroli tabeli zawartości wolumenu - VTOC Fixer. Jest on zasadniczo przeznaczony do kontroli tabeli zawartości dysku twardego, sformatowanego za pomocą MYDOS-a, lecz prawdopodobnie będzie również funkcjonował poprawnie, gdy zostanie użyty do kontroli tabeli zawartości dyskietki mającej format zgodny z DOS 2.0 i 2.5.

Działanie programu VTOC Fixer jest następujące: czyta on kolejno wszystkie pliki z określonego napędu; bada, które sektory zajmowane są przez czytany plik i na podstawie uzyskiwanych w ten sposób informacji tworzy nową tabelę zawartości wolumenu /VTOC/. Na koniec, utworzoną tabelę porównuje z tabelą zapisaną na dysku. Jeśli w wyniku porównania wykryje jakiegokolwiek rozbieżności, użytkownik zostanie o nich poinformowany i będzie mógł zlecić zapisanie poprawnej tabeli na dysk lub dyskietkę. W szczególności program VTOC Fixer sprawdza:

1. Czy poprawna jest nazwa pliku? Każda nazwa pliku zawierająca znaki w inwersie wideo, znaki uzyskiwane przez naciśnięcie klawisza CONTROL, zaczynająca się znakiem spacji lub cyfrą jest nazwą błędną.

2. Czy na dysku znajdują się błędne sektory? Błędny sektor znajdujący się w środkowej części powierzchni dysku (dotyczy to szczególnie dysku twardego) powoduje wygenerowanie przez DOS błędu podczas próby zapisu do niego informacji. W rezultacie nie można na dysk zapisać żadnego pliku. Problem ten można rozwiązać jedynie poprzez skopiowanie na dyskietki wszystkich plików z dysku twardego i ponowne jego sformatowanie. Jednak jest to metoda uciążliwa i pracochłonna, szczególnie gdy dysk twardy ma dużą pojemność i jest w znacznym stopniu wypełniony plikami danych. Program VTOC Fixer „proponuje” inne rozwiązanie, a mianowicie skasowanie pliku zawierającego uszkodzony sektor i wpisanie do tabeli zawartości wolumenu informacji wskazującej, że sektor ten jest zajęty. Po takiej operacji MYDOS nigdy nie będzie próbował użyć tego sektora.

3. Czy poprawne są bajty łącznikowe we wszystkich sektorach zajmowanych przez czytany plik? Bajty łącznikowe umieszczone są na końcu każdego sektora zajmowanego przez plik. Wskazują one numer następnego sektora, w którym zapisana jest treść pliku. Jeśli bajty łącznikowe zawierają numer sektora mniejszy od 4 lub większy do maksymalnej liczby sektorów zapisanych na dysku, to podczas próby odczytu pliku zawierającego taki sektor, sygnalizowany jest błąd nr 139. Wykrycie błędnych bajtów łącznikowych w sektorze VTOC Fixer sygnalizuje komunikatem „BAD RANGE”. VTOC Fixer skasuje plik, w którym wykryty zostanie sektor zawierający błędne bajty łącznikowe. Wszystkie sektory zajmowane przez skasowany plik oznaczone zostaną w tabeli VTOC jako wolne.

4. Czy na dysku znajdują się pliki, których bajty łącznikowe wskazują się nawzajem? Wykrycie takiego błędu sygnalizowane jest komunikatem „LOOP ERR”. VTOC Fixer skasuje każdy plik, w którym wykryty zostanie taki błąd, a sektory zajmowane przez skasowany plik zostaną oznaczone w tabeli zawartości wolumenu jako wolne.

5. Czy na dysku znajdują się pliki kolidujące ze sobą? Kolidacja plików ma miejsce, gdy np. sektor zajęty przez plik, katalog, podkatalog lub tabelę VTOC zostanie wykorzystany przez inny plik, katalog, lub podkatalog. Przyczyną kolidacji jest błąd w tabeli VTOC. Sektor, który w rzeczywistości jest zajęty, a w tabeli tej oznaczony jest jako wolny, może zostać użyty podczas zapisywania kolejnego pliku lub tworzenia podkatalogu. Wykrycie kolidacji plików sygnalizowane jest komunikatem „COLLISION”. VTOC Fixer skasuje kolidujące pliki, a zajmowane przez nie sektory oznaczone zostaną w tabeli zawartości wolumenu jako wolne.

6. Czy wszystkie pliki na dysku mają poprawną długość? Błędna długość pliku jest konsekwencją kolizji plików. Jeśli dwa pliki kolidują ze sobą to zajmują wspólnie jeden lub kilka sektorów. Ponieważ bajty łącznikowe wczytanego sektora zawierają numer następnego sektora, który trzeba odczytać, to odczytywanie sektorów pliku będącego w kolizji z innym plikiem może zakończyć się odczytaniem sektorów należących do drugiego z nich. W rezultacie liczba sektorów zajętych przez plik zapisana w katalogu może różnić się od liczby przeczytanych sektorów. Gdy przypadek taki zostanie wykryty przez program, to będzie wyprowadzony komunikat „BAD LENGTH”. VTOC Fixer skasuje pierwszy z plików będących w kolizji i oznaczy jako wolne wszystkie sektory nie zajmowane przez drugi z kolidujących ze sobą plików.

7. Czy poprawna jest wersja DOS-a? Każda, starsza (o numerze mniejszym od 4.50) wersja MYDOS-a zawiera błędy związane z formatowaniem dysków twardej. W wersjach tych podawana przez użytkownika liczba sektorów dostępnych na dysku nie jest zapamiętywana poprawnie. W rezultacie wyprowadzana przez te wersje liczba wolnych na dysku sektorów różni się od rzeczywistej o ok. 200 sektorów. VTOC Fixer sprawdza poprawność wersji DOS-a licząc liczbę wszystkich sektorów na dysku i opierając na niej wszystkie pozostałe kalkulacje.

JAK UŻYWAĆ PROGRAMU VTOC Fixer?

Program VTOC Fixer zapisany jest w pliku o nazwie VTOCFIX.COM. Po uruchomieniu go należy wprowadzić numer napędu dyskowego i nacisnąć klawisz RETURN. Dysk, w którym sprawdzana będzie poprawność tabeli zawartości wolumenu nie może być zabezpieczony przez zapisem.

Program po uruchomieniu wyświetla pytanie: „Verify files to delete?”. Udzielenie odpowiedzi „Y” spowoduje, że program nie wprowadzi żadnych zmian do sprawdzanego dysku bez uzyskania zgody użytkownika. Naciśnięcie „N” informuje program, że ewentualna korekta dysku i jego tabeli zawartości wolumenu ma być wykonana automatycznie bez interwencji użytkownika.

Następnie program oblicza liczbę wszystkich sektorów dostępnych na dysku. Liczba ta zostanie wyświetlona. Jeśli użytkownik z jakichkolwiek przyczyn jej nie zaakceptuje, to przez naciśnięcie klawisza „N” może zasygnalizować chęć jej zmiany. Po wprowadzeniu przez użytkownika nowej liczby sektorów VTOC Fixer próbuje odczytać sektor o najwyższym numerze. Jeśli wynik próby będzie negatywny (liczba sektorów na dysku jest mniejsza od podanej) należy ponownie określić liczbę sektorów. Proces ten będzie trwał tak długo, aż podana liczba sektorów zostanie przez program zaakceptowana.

Proces kontroli poprawności tabeli VTOC trwa dosyć długo. Podczas jego realizacji program VTOC Fixer wyświetla nazwę aktualnie badanego pliku (bez kropki poprzedzającej rozszerzenie) i komunikaty o wykrytych błędach.

Uwaga! Jeśli po pewnym czasie na ekranie zacznie się cyklicznie zmieniać kolor tła i liter, to przywrócenie normalnego trybu wyświetlania nastąpi po naciśnięciu SHIFT CONTROL A.

Program VTOC Fixer wyświetla następujące komunikaty o wykrytych błędach:

COLLISION - badany plik zajmuje sektor, który zajęty jest również przez inny plik, podkatalog lub tablicę VTOC.

BAD LENGTH - długość pliku zapisana w katalogu nie odpowiada jego rzeczywistej długości. Jest to prawdopodobnie rezultat kolizji plików.

LOOP ERR - badany plik zawiera sektory o bajtach łącznikowych wzajemnie się wskazujących.

BAD RANGE - bajty łącznikowe sektora wskazują na sektor o numerze mniejszym od 4 lub większym od liczby dostępnych sektorów na dysku.

Err #144 @ Sec #[numer sektora] - informuje o wykryciu na dysku uszkodzonego sektora.

INVALID FILENAME:[nazwa pliku] - informuje o wykryciu niepoprawnej nazwy pliku.

Każdy plik, w którym zostanie wykryta jakakolwiek nieprawidłowość powinien być skasowany przez program VTOC Fixer. Jeśli nie zostanie skasowany, to i tak zajmowane przez niego sektory zostaną w przyszłości użyte przez jakikolwiek plik zapisywany na dysku.

Na koniec VTOC Fixer wyświetla liczbę sektorów zajętych przez pliki, liczbę sektorów wolnych na dysku obliczoną na podstawie wygenerowanej tablicy zawartości wolumenu, liczbę wolnych sektorów odczytaną z kontrolowanej tablicy oraz korektę liczby wolnych sektorów, podaną podczas kontroli poprawności wersji MYDOS-a. Jeśli liczba wolnych sektorów zapisana w tabeli VTOC jest zgodna z liczbą wyznaczoną przez program, to dysk nie wymaga żadnej korekty. W przeciwnym wypadku program zapyta, czy zapisać na dysk poprawną tabelę zawartości wolumenu. Na pytanie to należy odpowiedzieć twierdząco „Y”, w celu naprawienia błędów. Jeśli użytkownik z jakichkolwiek powodów nie chce, aby zbudowana przez program tabela VTOC została zapisana na dysk, może nacisnąć „N” i RETURN.

* * *

OD AUTORA KONWERSJI

Dokonując „rewitalizacji” tego dokumentu stanąłem przed dylematem: czy zrobić to słowo w słowo czy też wprowadzić własne poprawki. O ile korekta błędów wydaje się oczywista, to inne zmiany już niekoniecznie. Innymi słowy: czy zachować archiwalny charakter dokumentu, czy go „unowocześnić”? Przecież od początku lat 90. XX wieku, kiedy ten dokument został opracowany, nomenklatura informatyczna (przynajmniej w języku polskim) uległa pewnym zmianom. W tekście też pojawiają się niezręczności czy niejasności językowe, które, jak sądzę, są wynikiem pośpiechu w tłumaczeniu, a utrudniające zrozumienie tekstu. Ostatecznie uznałem, że ma to być „żywy” dokument służący zainteresowanym w poznaniu MyDOS-a, a nie wyłącznie archiwalna ciekawostka.

Jednakże starałem się ograniczyć zakres zmian do, moim zdaniem, absolutnego minimum. Dlatego pozostawiłem sformułowania typu „Czy użyć standardowych parametrów do utworzenia...”. Dziś użylibyśmy „Czy użyć domyślnych parametrów...”. Niektóre zmiany polegają tylko na przestawieniu szyku słów w zdaniu, co ułatwiło jego zrozumienie. Nieco więcej zmian zawiera tabela błędów na str. 28; sądzę, że moja interpretacja błędów nr 129, 131, 135, 143, 163, 173 i 181 lepiej objaśnia problem.

Mając nadzieję, iż nie przyczyniłem się do powstania (przynajmniej zbyt wielu) nowych błędów zapraszam do lektury.

Bluki

Pomocnicze materiały źródłowe:

1. *MYDOS Version 4 User Guide. Revision 4.50 for Atari Home Computers.* Copyright (C) 1988 by WORDMARK Systems and the authors: Charles Marslett and Robert Puff.
2. *MYDOS Version 4 Technical User Guide. Revision 4.50 for Atari Home Computers.* Copyright (C) 1988 by WORDMARK Systems and the authors: Charles Marslett and Robert Puff.
3. *Documentation for The VTOC Fixer!* By Robert Puff 11/27/88.