



Windows 7

Komendy i polecenia

Praktyczne przykłady

- Poznaj możliwości oferowane przez tryb tekstowy
- Dowiedz się, jak tworzyć pliki wsadowe
- Naucz się automatyzować działanie systemu

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiegokolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicieli.

Autor oraz Wydawnictwo HELION dołożyli wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie biorą jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Autor oraz Wydawnictwo HELION nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Redaktor prowadzący: Michał Mrowiec

Projekt okładki: Jan Paluch

Fotografia na okładce została wykorzystana za zgodą Shutterstock.com.

Wydawnictwo HELION
ul. Kościuszki 1c, 44-100 GLIWICE
tel. 32 231 22 19, 32 230 98 63
e-mail: helion@helion.pl
WWW: <http://helion.pl> (księgarnia internetowa, katalog książek)

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

<http://helion.pl/user/opinie?wi7kpp>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Kody źródłowe wybranych przykładów dostępne są pod adresem:

<ftp://ftp.helion.pl/przyklady/wi7kpp.zip>

ISBN: 978-83-246-3537-5

Copyright © Helion 2011

Printed in Poland.

- [Kup książkę](#)
- [Poleć książkę](#)
- [Oceń książkę](#)

- [Księgarnia internetowa](#)
- [Lubię to! » Nasza społeczność](#)

Spis treści

Wstęp	9
W.1. Zalety trybu tekstowego	10
W.2. Korzyści wynikające ze stosowania programów wsadowych	11
W.3. Twój podręcznik programowania	12
Rozdział 1. System operacyjny — FAQ	13
1.1. Co oznaczają skróty DOS i MS-DOS?	13
1.2. Z jakich elementów składa się system komputerowy?	13
1.3. Co to jest system operacyjny?	14
1.4. Na czym polega przetwarzanie wsadowe?	15
1.5. Co to jest program wsadowy?	15
1.6. Co to jest spooling?	16
1.7. Na czym polega praca z podziałem czasu?	16
1.8. Co to jest swapping?	16
1.9. Kiedy komputery mogą komunikować się ze sobą?	17
1.10. Co to jest protokół sieciowy?	17
Rozdział 2. Tryb tekstowy	19
2.1. Jak pracować w trybie konsoli?	20
2.2. Jak pracować z programami wsadowymi?	22
cmd	22
Znaki cudzysłowu w poleceniu	27
2.3. Co można uzyskać, włączając rozszerzenia poleceń?	28
Opóźnione rozwijanie zmiennych środowiskowych	30
Ścieżki relatywne i bezwzględne	30
Dokończanie nazw plików i katalogów	32
Skróty klawiaturowe	34
Dodatkowe informacje o poleceniu	34
Rozdział 3. Operacje na katalogach	35
3.1. Jak zakładać katalogi?	35
md	35
mkdir	50
3.2. Jak zmieniać nazwy katalogów?	50
move	50
3.3. Jak usuwać katalogi?	55
rd	55

3.4.	Jak zmieniać katalogi?	59
	cd	59
	chdir	63
3.5.	Jak wyświetlać graficzną strukturę katalogów?	63
	tree	63
3.6.	Jak kopiować drzewo katalogów?	66
	xcopy	66
Rozdział 4.	Operacje na plikach	73
4.1.	Jak wyświetlać zawartość pliku tekstowego?	74
	type	74
4.2.	Jak wyszukiwać ciągi znaków?	77
	find	77
	findstr	81
4.3.	Jak sortować wiersze w pliku?	85
	sort	85
4.4.	Jak drukować pliki tekstowe?	87
	print	87
4.5.	Jak przenosić pliki?	88
	move	88
4.6.	Jak usuwać pliki?	88
	del	89
	erase	92
4.7.	Jak wyświetlić listę plików i katalogów?	92
	dir	92
4.8.	Jak zmieniać rozszerzenia nazw plików?	96
	assoc	96
	ftype	97
4.9.	Jak zmieniać atrybuty plików?	98
	attrib	98
4.10.	Jak ustawiać listę kontroli dostępu do plików?	100
	icacls	100
4.11.	Jak porównywać pliki?	106
	comp	106
	fc	107
4.12.	Jak kopiować pliki?	109
	copy	109
4.13.	Jak zmieniać nazwy plików?	112
	ren	112
	rename	113
4.14.	Jak zamieniać pliki?	113
	replace	113
4.15.	Jak porównywać zawartość dyskiek?	114
	diskcomp	114
4.16.	Jak kopiować zawartość dyskiek?	115
	diskcopy	115
4.17.	Jak rozpakowywać pliki instalacyjne?	116
	expand	116
Rozdział 5.	Zarządzanie zadaniami	117
5.1.	Jak uruchamiać polecenia w określonym czasie?	117
	at	117
	schtasks	122

5.2.	Jak wywoływać programy wsadowe?	132
	call	132
5.3.	Jak wstrzymywać programy wsadowe?	136
	pause	136
5.4.	Jak wyłączać przerywanie wykonywania zadań?	138
	break	138
5.5.	Jak wyświetlać nowe okno?	138
	start	138
5.6.	Jak zmienić tytuł okna wiersza poleceń?	141
	title	141
5.7.	Jak wyczyścić okno wiersza poleceń?	142
	cls	142
5.8.	Jak wykonywać zadania w pętli?	143
	for	143
5.9.	Jak przejść do instrukcji oznaczonej etykietą?	146
	goto	146
5.10.	Jak napisać instrukcję warunkową?	147
	if	147
5.11.	Jak sterować wyświetlaniem informacji na ekranie?	150
	more	150
5.12.	Jak zachować bieżący katalog dla polecenia POPD?	153
	pushd	153
5.13.	Jak przejść do katalogu zachowanego przez polecenie PUSHD?	153
	popd	153
5.14.	Jak definiować zmienne środowiskowe dla plików wsadowych?	154
	setlocal	154
5.15.	Jak zmienić położenie parametrów w pliku wsadowym?	157
	shift	157
5.16.	Co to jest host skryptów systemu Windows?	159
	cscript	159
5.17.	Jak kończyć pracę skryptów wsadowych?	161
	exit	161
5.18.	Jak wyszukiwać i odłączać otwarte pliki?	162
	openfiles	162
	tasklist	165
	taskkill	166
5.19.	Jak zamknąć system?	167
	shutdown	167
Rozdział 6.	Środowisko	169
6.1.	Jak odczytać wersję systemu operacyjnego?	169
	ver	169
6.2.	Jak zdefiniować tekst zgłoszenia trybu konsoli?	170
	prompt	170
6.3.	Jak ustawić ścieżkę wyszukiwania dla plików wykonywalnych?	172
	path	172
6.4.	Jak ustawiać kolor tła i napisów konsoli?	173
	color	173
6.5.	Jak ustawić stronę kodową?	175
	chcp	175
6.6.	Jak ustawić czas i datę systemu operacyjnego?	176
	time	176
	date	177

6.7.	Jakie są dodatkowe funkcje wiersza poleceń?	178
	doskey	178
6.8.	Jak wyświetlać komunikaty ekranowe?	183
	echo	183
6.9.	Jak uzyskiwać informacje o poleceniach systemu Windows?	185
	help	185
6.10.	Jak konfigurować urządzenia systemowe?	187
	mode	187
6.11.	Jak definiować zmienne środowiskowe?	191
	set	191
	Podstawianie zmiennej środowiskowej	193
	Obsługa opóźnionego rozwijania zmiennych środowiskowych	194
	Zmienne środowiskowe	194
6.12.	Jak włączyć dysk wirtualny?	195
	subst	195
6.13.	Jak wyświetlić listę sterowników urządzeń?	196
	driverquery	196

Rozdział 7. System plików 199

7.1.	Jak formatować nośniki?	199
	format	199
7.2.	Jak montować woluminy?	201
	mountvol	201
7.3.	Jak odczytać dane woluminu?	203
	vol	203
7.4.	Jak ustawić etykietę woluminu lub dysku?	203
	label	203
7.5.	Jak kontrolować poprawność zapisywania plików na dysku?	204
	verify	204
7.6.	Jak defragmentować woluminy?	205
	defrag	205
7.7.	Jak testować i naprawiać nośniki?	206
	chkdsk	206
	chkntfs	208
7.8.	Jak odzyskiwać dane z uszkodzonego dysku?	210
	recover	210
7.9.	Jak kompresować dane na partycjach NTFS?	210
	compact	210
7.10.	Jak konwertować system FAT na NTFS?	212
	convert	212
7.11.	Jak szyfrować dane na partycjach NTFS?	214
	cipher	214
7.12.	Jak mierzyć wydajność dysków?	217
	diskperf	217
7.13.	Zarządzanie przydziałami dysku z wiersza polecenia	218
	fsutil	218

Rozdział 8. Praca w sieci 223

8.1.	Jak odczytać adres MAC karty sieciowej?	223
	getmac	223
8.2.	Jak skonfigurować kartę sieciową?	226
	ipconfig	226
8.3.	Jak usprawnić pracę w kilku sieciach?	229

8.4.	Jak ustawić parametry połączenia sieciowego?	231
	net	231
8.5.	Jak testować połączenie sieciowe?	257
	ping	257
8.6.	Jak odczytać statystykę połączenia TCP/IP?	258
	netstat	258
8.7.	Jak wysyłać komunikaty do użytkowników?	260
	msg	260
8.8.	Jak pracować z terminalem?	261
	tscon	261
8.9.	Jak zbierać informacje o systemie?	261
	systeminfo	261
8.10.	Statystyki protokołu i bieżące połączenia TCP/IP	262
	nbtstat	262
8.11.	Modyfikowanie konfiguracji sieciowej uruchomionego komputera	263
	netsh	263
	Polecenia netsh dotyczące usługi DHCP	265
	Jak zresetować protokół TCP/IP?	267
	Jak przypisać statyczne dane protokołu TCP/IP?	267
Dodatek A	Parametry wsadowe, zmienne i filtry	269
A.1.	Parametry wsadowe	269
A.2.	Modyfikatory	271
A.3.	Stosowanie wielu poleceń i symboli przetwarzania warunkowego	272
A.4.	Systemowe i lokalne zmienne środowiskowe	273
A.5.	Ustawianie zmiennych środowiskowych	276
A.6.	Podstawianie wartości zmiennych środowiskowych	277
A.7.	Filtry	277
	Polecenie more	277
	Polecenie find	280
	Polecenie sort	281
	Skorowidz	283

Rozdział 3.

Operacje na katalogach

Katalogi można porównać do segregatorów w biurze, a pliki do dokumentów. Aby zaprowadzić porządek w biurze, należy dokumenty pogrupować i wpiąć do segregatorów. W przypadku plików trzeba utworzyć strukturę katalogów ułatwiającą zlokalizowanie danego pliku i zapisywać pliki w przeznaczonych do tego celu miejscach.

3.1. Jak zakładać katalogi?

Katalogi można zakładać za pomocą dwóch poleceń: `md` oraz `mkdir`.

md

Polecenie `md` umożliwia założenie katalogu.

Ma ono następującą składnię:

`MD [dysk:]ścieżka`

gdzie:

dysk — litera napędu;

ścieżka — ścieżka dostępu do zakładanego katalogu.

Jak utworzyć folder w katalogu bieżącym?

Poznanie polecenia `md` rozpoczniemy od operacji najprostszej. Utworzymy folder w katalogu bieżącym.

Przykład 3.1.

Utwórz na dysku bieżącym folder *a*.

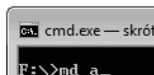
Wpisz w oknie konsoli polecenie

```
md a
```

Na rysunku 3.1 pokazano polecenie wpisane w oknie konsoli.

Rysunek 3.1.

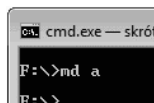
*Polecenie wpisane
w oknie konsoli*



Naciśnij klawisz *Enter* (rysunek 3.2).

Rysunek 3.2.

*Polecenie zostało
wykonane*



Jak sprawdzić rezultat wykonania polecenia?

Tym, co najbardziej mi przeszkadza podczas pracy w środowisku tekstowym, jest poruszanie się po omacku.

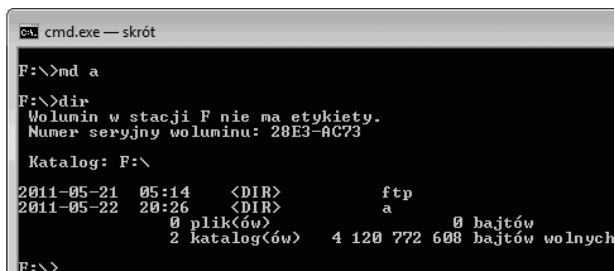
Skąd wiadomo, że po wykonaniu poleceń pokazanych na rysunku 3.2 utworzony zostanie folder o nazwie *a*?

Czy na partycji *F* jest jeden folder, czy jest ich wiele?

Na szczęście istnieją polecenia trybu tekstowego, które pozwalają na wyświetlenie zawartości folderu. Są to *dir* (rysunek 3.3) oraz *tree* (rysunek 3.4).

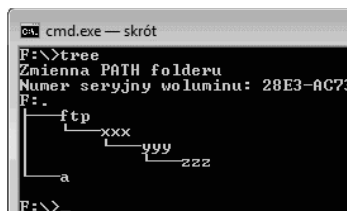
Rysunek 3.3.

*Wynik wykonania
polecenia dir*



Rysunek 3.4.

*Wynik wykonania
polecenia tree*



Jeżeli w folderze znajduje się wiele plików i podfolderów, informacja o nich nie zmieści się w oknie konsoli. Zawartość okna można przewinąć.

Polecenie `dir` wywołane bez parametrów powoduje wyświetlenie listy plików i podkatalogów katalogu. Nie dostarcza ono jednak informacji o złożoności struktury. Nie wiemy, ile podkatalogów zawierają katalogi.

Moim zdaniem do zobrazowania struktury katalogów znacznie lepiej nadaje się polecenie `tree`. Przedstawia ono graficzną strukturę plików i folderów na dysku lub w określonej ścieżce. Dzięki temu na rysunku 3.4 widać, że folder *a* nie zawiera podkatalogów, zaś folder *ftp* ma ich trzy.

Ale o czym to mówiliśmy? Aha, wiemy już, że w wyniku wykonania poleceń pokazanych na rysunku 3.2 folder *a* został utworzony.



Wskazówka

Polecenia `dir` i `tree` mają wiele opcji. Opis ich znajduje się w dalszej części książki. W tym miejscu wspomniałem tylko o istnieniu poleceń, które pozwalają sprawdzać rezultaty wykonania innych poleceń.

Skoro wiemy, jak sprawdzić zawartość folderu, możemy powrócić do zakładania folderów.

Jak utworzyć folder we wskazanej lokalizacji?

Nie zawsze zachodzi taki szczęśliwy zbieg okoliczności, że folderem bieżącym jest ten, w którym chcemy utworzyć podkatalog. Można oczywiście wydać polecenie zmiany katalogu. Prostszy rozwiązaniem jest użycie w charakterze argumentu polecenia ścieżki do tworzonego katalogu.

Przykład 3.2.

Utwórz folder *b* na dysku *F*.

Wpisz w oknie konsoli polecenie:

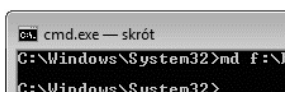
```
md f:\b
```

Naciśnij klawisz *Enter*.

Polecenie zostanie wykonane (rysunek 3.5).

Rysunek 3.5.

W oknie konsoli
nie widać wyniku
wykonania polecenia



Czy system sygnalizuje błąd podczas ponownej próby tworzenia już istniejącego katalogu?

Wiemy już, jak w oknie konsoli wyświetlić zawartość katalogu (rysunki 3.3 i 3.4). Fakt utworzenia folderu wykorzystamy do sprawdzenia, czy możliwe jest nadpisanie go.

Przykład 3.3.

Wykonaj polecenia opisane w przykładzie 3.2.

Wpisz w oknie konsoli polecenie:

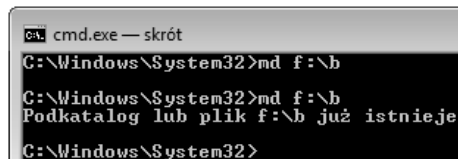
```
md f:\b
```

Naciśnij klawisz *Enter*.

Folder nie zostanie utworzony. Wyświetlona zostanie informacja o przyczynie niewykonania polecenia (rysunek 3.6).

Rysunek 3.6.

Nie jest możliwe nadpisanie folderu



```
C:\> cmd.exe — skrót
C:\Windows\System32>md f:\b
C:\Windows\System32>md f:\b
Podkatalog lub plik f:\b już istnieje.
C:\Windows\System32>
```

Czy system zasygnalizuje błąd podczas próby utworzenia drzewa katalogów w istniejącym folderze?

W przykładzie 3.3. nie udało się utworzyć folderu o nazwie identycznej z już istniejącym. Sprawdźmy, czy uda się utworzyć nie jeden folder, ale całą strukturę.

Przykład 3.4.

Na dysku o nazwie *F* utwórz folder *b*. W folderze *b* utwórz folder *c*. W folderze *c* utwórz folder *d*. W folderze *d* utwórz folder *e*.

Jak wykonać taką operację, posługując się Eksploratorem Windows? Należy wyświetlić dysk *F*, utworzyć folder *b*, wejść do folderu *b*, utworzyć folder *c*, wejść do folderu *c*... Zadanie jest mozolne.

Wpisz w oknie konsoli polecenie:

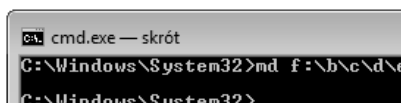
```
md f:\b\c\d\e
```

Naciśnij klawisz *Enter*.

Polecenie zostanie wykonane (rysunek 3.7).

Rysunek 3.7.

W tym przypadku istnienie folderu b nie było przeszkodą w wykonaniu polecenia!



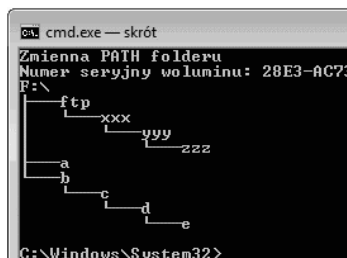
```
C:\> cmd.exe — skrót
C:\Windows\System32>md f:\b\c\d\e
C:\Windows\System32>
```

Zapisanie w jednej linii polecenia umożliwiającego utworzenie wielu folderów jest możliwe przy włączonych rozszerzeniach poleceń — domyślna opcja pracy konsoli.

Ostatni przykład dobitnie pokazuje, że możliwości, jakie stwarzają polecenia dostępne w trybie tekstowym, są większe niż oferowane przez interfejs graficzny. Poleceniem wpisanym w jednej linii można utworzyć wiele katalogów (rysunek 3.8). Osiągnięcie tego samego rezultatu za pomocą posługiwania się „okienkami” jest bardziej pracochłonne.

Rysunek 3.8.

Drzewo folderów utworzone przez wykonanie polecenia z rysunku 3.7



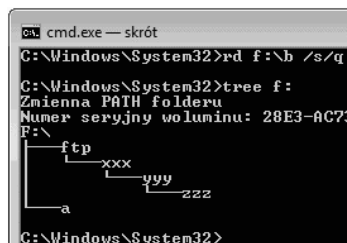
```
cmd.exe — skrót
Zmienna PATH folderu
Numer seryjny woluminu: 28E3-AC73
F:\
├── ftp
│   ├── xxx
│   └── yyy
│       └── zzz
├── a
├── b
│   └── c
│       └── d
│           └── e
```

Czy w trybie tekstowym można usunąć folder?

Do wykonania kolejnego przykładu potrzebne będzie usunięcie drzewa folderów (rysunek 3.8). Jak przystało na użytkowników trybu tekstowego, nie będziemy uciekali się do pracy z Eksploratorem Windows. Wykorzystamy polecenie `rd` (rysunek 3.9). Szerzej zostało ono opisane w dalszej części książki.

Rysunek 3.9.

Drzewo folderów zostało usunięte skutecznie i bez pytania o pozwolenie



```
cmd.exe — skrót
C:\Windows\System32>rd f:\b /s/q
C:\Windows\System32>tree f:
Zmienna PATH folderu
Numer seryjny woluminu: 28E3-AC73
F:\
├── ftp
│   ├── xxx
│   └── yyy
│       └── zzz
└── a
```



Pliki usunięte za pomocą polecenia wydanego w trybie tekstowym nie są przenoszone do Kosza, a usuwane bezpośrednio.

Jaki wpływ ma wyłączenie rozszerzeń poleceń na możliwości tworzenia drzewa folderów?

Po wyłączeniu rozszerzeń poleceń nie można w jednym wierszu zapisać polecenia utworzenia więcej niż jednego katalogu.

Przykład 3.5.

Uruchom okno konsoli z wyłączonymi rozszerzeniami poleceń. Utwórz na dysku o nazwie *F* folder *b*. W folderze *b* utwórz folder *c*. W folderze *c* utwórz folder *d*. W folderze *d* utwórz folder *e*.



Wskazówka

Do uruchomienia konsoli z wyłączonymi rozszerzeniami poleceń wykorzystaj plik wsadowy *bezrozs.bat*.

```
:: bezrozs.bat
:: Program uruchamia konsolę z wyłączonymi rozszerzeniami poleceń.
cmd /e:off
```

Znajduje się on na ftp wydawnictwa Helion.

1. Uruchom okno konsoli, korzystając z pliku *bezrozs.bat*.
2. Wpisz w oknie konsoli polecenie `md f:\b\c\d\e`.
3. Naciśnij klawisz *Enter*.
4. Polecenie nie zostanie wykonane (rysunek 3.10). Powodem tego są wyłączone rozszerzenia.

Rysunek 3.10.

Po wyłączeniu rozszerzeń utworzenie struktury podkatalogów jednym poleceniem jest niewykonalne

```
C:\Windows\system32\cmd.exe
F:\ftp>cmd /e:off
Microsoft Windows [Wersja 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. Wszelkie prawa zastrzeżone.

F:\ftp>md f:\b\c\d\e
Nie można utworzyć katalogu lub pliku.

F:\ftp>
```

5. Identyczną strukturę katalogów, jaką chcieliśmy utworzyć w punkcie 2., można założyć za pomocą pojedynczych poleceń. Wpisz polecenie `md f:\b`. Utworzony zostanie katalog *b*.
6. Przejdź do katalogu *b*, wpisując polecenie `cd b`.
7. Wpisz polecenie `md c`. Utworzony zostanie katalog *c*.
8. Przejdź do katalogu *c*, wpisując polecenie `cd c`.
9. Wpisz polecenie `md d`. Utworzony zostanie katalog *d*.
10. Przejdź do katalogu *d*, wpisując polecenie `cd d`.
11. Wpisz polecenie `md e`. Utworzony zostanie katalog *e*.
12. Przejdź do katalogu *e*, wpisując polecenie `cd e`. Uzyskaliśmy taką samą strukturę katalogów jak w przykładzie 3.4. Wymagało to jednak wpisania dużo większej liczby poleceń (rysunek 3.11).

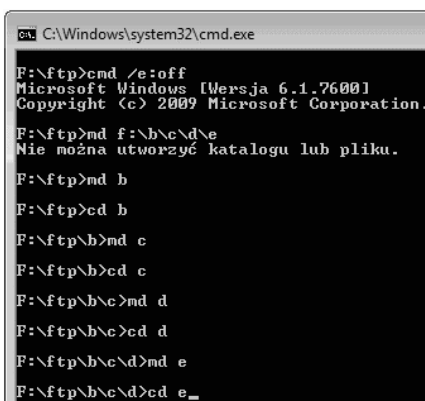
Jak program wsadowy może ułatwić życie?

Jeśli wykonałeś polecenia podane w przykładzie 3.5, wiesz już, jak żmudne może być wpisywanie poleceń z klawiatury. A ile przy tej okazji można popełnić pomyłek!

Komputer wyśmienicie nadaje się do wykonywania czynności powtarzalnych. Nie myli się przy tym i nie męczy. Czy jest pracownikiem idealnym? Zdecydowanie NIE! Komputer również nie myśli. Wykonuje idealnie tylko to, co zlecił mu programista.

Rysunek 3.11.

Po wyłączeniu
rozszerzeń drzewo
katalogów trzeba
tworzyć krok po kroku



```
C:\Windows\system32\cmd.exe
F:\ftp>cmd /e:off
Microsoft Windows [Wersja 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.

F:\ftp>md f:\b\c\d\e
Nie można utworzyć katalogu lub pliku.

F:\ftp>md b
F:\ftp>cd b
F:\ftp\b>md c
F:\ftp\b>cd c
F:\ftp\b\c>md d
F:\ftp\b\c>cd d
F:\ftp\b\c\d>md e
F:\ftp\b\c\d>cd e_
```

Pamiętając o tych ograniczeniach, można wiele prac z komputerem znacznie uprościć. Wystarczy zauważyć, że coś można zapisać w postaci programu, i napisać program.

W takiej sytuacji jak w przykładzie 3.5 znacznym ułatwieniem może być program pozwalający na wpisanie tylko kolejnych nazw folderów.

Przykład 3.6.

Napisz program, który będzie zakładał drzewo zbudowane z czterech folderów.

Polecenie jest wykonywane przez program:

```
:: utwkat.bat
:: Program zakłada drzewo zbudowane z czterech folderów.
:: Nazwy folderów należy wprowadzić jako parametry programu.
@ECHO OFF
md %1
cd %1
md %2
cd %2
md %3
cd %3
md %4
cd %4
```

Pora na wyjaśnienie, co i jak wykonuje program.



Wewnątrz programu wsadowego argumenty jego wywołania reprezentowane są przez napis `%n`, gdzie `n` oznacza numer argumentu.

Pierwsze trzy linie rozpoczynają się od dwukropka. Oznacza on, że wszystko, co znajduje się na prawo od niego, powinno być traktowane jako komentarz. Nie jest to instrukcja dla komputera, ale informacja dla człowieka, który będzie używał programu. Umieszczanie komentarzy w programach jest dobrym zwyczajem, gdyż zamiast śledzić treść programu, by się dowiedzieć, do czego on służy — wystarczy przeczytać komentarz. Drugi dwukropek ułatwia odnalezienie komentarza w długim programie. Po dwukropkach wpisana jest nazwa programu.

Linia

```
md %1
```

powoduje utworzenie folderu o nazwie identycznej z pierwszym parametrem wywołania polecenia.

Linia

```
vd %1
```

powoduje przejście do folderu o nazwie identycznej z pierwszym parametrem wywołania polecenia.

Linia

```
md %2
```

powoduje utworzenie folderu o nazwie identycznej z drugim parametrem wywołania polecenia.

Linia

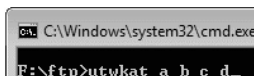
```
cd %2
```

powoduje przejście do folderu o nazwie identycznej z drugim parametrem wywołania polecenia.

Roli reszty linii łatwo się domyślić.

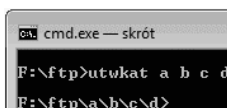
Skoro w książce mają znajdować się praktyczne przykłady, należy program przetestować w praktyce.

1. Uruchom okno konsoli.
2. Przejdź do folderu, w którym znajduje się plik *utwkat.bat*.
3. Wpisz w oknie konsoli *utwkat a b c d*. Litery *a b c d* są parametrami wywołania programu (rysunek 3.12).



Rysunek 3.12. Parametry wywołania programu mają postać ciągu znaków wpisanych po nazwie programu

4. Naciśnij klawisz *Enter*.
5. Polecenie zostanie wykonane (rysunek 3.13). Na nośniku utworzona zostanie struktura katalogów (rysunek 3.14).



Rysunek 3.13. Parametry wywołania programu mają postać ciągu znaków wpisanych po nazwie programu

Rysunek 3.14.

*Drzewo folderów
zostało utworzone*

```

cmd.exe — skrót
F:\>tree
Zmienna PATH folderu
Numer seryjny woluminu: 28E3-AC73
F:.
├── ftp
│   ├── a
│   │   ├── b
│   │   │   ├── c
│   │   │   └── d
│   └── a
└── a
F:\>

```

6. Czy program jest wspaniący? Uprościł pracę, ale jest mało elastyczny. Co się stanie, gdy drzewo katalogów będzie miało się składać nie z czterech, a trzech elementów? Uruchom program z parametrami *x y z* (rysunek 3.15).

Rysunek 3.15.

*Drzewo folderów
zostało utworzone,
ale wyświetlony został
komunikat o błędzie*

```

cmd.exe — skrót
F:\ftp>utwkat x y z
Składnia polecenia jest niepoprawna.
F:\ftp>x\y\z
F:\ftp\x\y\z>cd\

```

7. Bez modyfikowania treści programu nie uda się również utworzyć drzewa składającego się z więcej niż czterech katalogów (rysunek 3.16).

Rysunek 3.16.

*Program utwkat
nie założy więcej
niż cztery katalogi*

```

cmd.exe — skrót
F:\ftp>utwkat 1 2 3 4 5 6
F:\ftp\1\2\3\4>

```

Przykład 3.7.

Napisz program, który będzie zakładał dowolną liczbę folderów. Foldery będą nosiły nazwy kolejnych liczb naturalnych.

Polecenie jest wykonywane przez program:

```

:: utwkat1.bat
:: Program zakłada n folderów o nazwach identycznych jak liczby
:: naturalne.
@ECHO OFF

for /L %%1 in (1,1,5) do md %%1

```

Pora na wyjaśnienie, co i jak wykonuje program.

Pierwsze trzy linie to komentarz.

Użyta została pętla FOR. Szerzej zostanie ona omówiona w rozdziale dotyczącym zarządzania zadaniami. Teraz wyjaśnię tylko to, co jest niezbędne do wykonania polecenia.

Pętla FOR ma następującą składnię:

```
FOR /L %zmienna IN(start,krok,koniec) DO polecenie [parametry_polecenia]
```

gdzie:

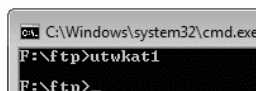
- /L* — opcja powoduje iterację ciągu numerycznego;
- zmienna* — określa jednoliterowy parametr wymienny;
- start* — parametr użyty do inicjowania skryptu;
- krok* — wartość, o którą zwiększany jest parametr za każdym wykonaniem pętli;
- koniec* — wartość, po osiągnięciu której pętla kończy działanie;
- polecenie [parametry_polecenia]* — polecenie wykonywane w pętli.

Zatem pętla generuje ciąg liczb od wartości *start* do wartości *koniec*. Zmieniają się one o wartość *krok*. Tak więc (1,1,5) generuje ciąg 1 2 3 4 5.

Skoro w książce mają znajdować się praktyczne przykłady, należy program przetestować w praktyce:

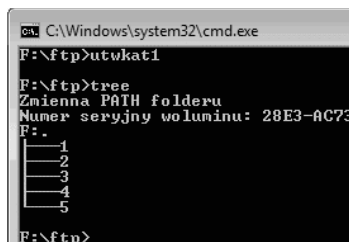
1. Uruchom okno konsoli.
2. Przejdź do folderu, w którym znajduje się plik *utwkat1.bat*.
3. Wpisz w oknie konsoli *utwkat1*.
4. Naciśnij klawisz *Enter*.
5. Polecenie zostanie wykonane (rysunek 3.17).

Rysunek 3.17.
Polecenie zostało wykonane



6. Na nośniku utworzona zostanie struktura katalogów (rysunek 3.18). W przykładzie utworzonych zostało pięć folderów. Nic nie stoi na przeszkodzie, aby folderów było znacznie więcej. Wystarczy tylko wpisać odpowiednią liczbę w programie i wykonać go.

Rysunek 3.18.
Utworzone zostało pięć folderów o nazwach odpowiadających kolejnym liczbom



Przykład 3.8

Napisz program, który będzie zakładał folder o nazwie identycznej z zapisem bieżącej daty.

Polecenie jest wykonywane przez program:

```
:: datedir.bat  
:: Program zakłada katalog o nazwie identycznej jak bieżąca data.  
@ECHO OFF  
MD %DATE%
```

Pora na wyjaśnienie, co i jak wykonuje program.

Pierwsza linia rozpoczyna się od dwukropka. Oznacza on, że wszystko, co znajduje się na prawo od niego, powinno być traktowane jak komentarz. Druga linia również zawiera komentarz. Opisana jest w nim funkcja, którą realizuje program.

Linia

```
@ECHO OFF
```

rozpoczyna się od znaku @. Zapobiega on wyświetleniu na początku polecenia pojedynczego wiersza z pliku wsadowego.

Polecenie `echo off` powoduje wyłączenie wyświetlania na ekranie poleceń wydawanych komputerowi przez program.

Linia

```
MD %DATE%
```

zawiera polecenie założenia folderu o nazwie identycznej ze zmienną `%DATE%`. Zmiennej `%DATE%` przypisywana jest bieżąca data. Folder zakładany jest w katalogu bieżącym.

Przy włączonych rozszerzeniach poleceń istnieje kilka dynamicznych zmiennych środowiskowych, które są obliczane dynamicznie za każdym razem, gdy wartość zmiennej jest rozwijana:

- `%CD%` — powoduje przypisanie zmiennej ciągu katalogu bieżącego,
- `%DATE%` — powoduje przypisanie zmiennej aktualnej daty,
- `%TIME%` — powoduje przypisanie zmiennej bieżącego czasu,
- `%RANDOM%` — powoduje przypisanie zmiennej losowej liczby dziesiętnej z przedziału od 0 do 32767,
- `%ERRORLEVEL%` — powoduje przypisanie zmiennej wartości `ERRORLEVEL`,
- `%CMDEXTPVERSION%` — powoduje przypisanie zmiennej wersji rozszerzeń procesora poleceń,
- `%CMDCMDLINE%` — powoduje przypisanie zmiennej oryginalnego wiersza polecenia wywołanego przez procesor poleceń.



Więcej informacji na temat polecenia `set` możesz uzyskać, wpisując w oknie konsoli polecenie `set /?`.



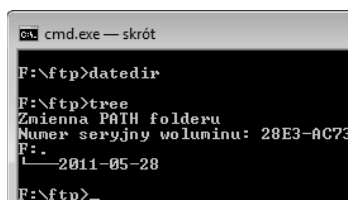
Kolejne uruchomienie programu *datedir.bat* nie powoduje zastąpienia istniejącego już folderu.

Skoro w książce mają znajdować się praktyczne przykłady, należy program przetestować w praktyce:

1. Uruchom okno konsoli.
2. Przejdź do folderu, w którym znajduje się plik *datedir.bat*.
3. Wpisz w oknie konsoli *datedir.bat*.
4. Naciśnij klawisz *Enter*.
5. Polecenie zostanie wykonane (rysunek 3.19).

Rysunek 3.19.

Polecenie zostało wykonane



Przykład 3.9.

Napisz program, który będzie zakładał folder o nazwie podanej po wyświetleniu pytania.

Polecenie jest wykonywane przez program:

```

:: kat.bat
:: Program zakłada katalog o nazwie wpisanej w odpowiedzi na pytanie.

@echo off
set /P kat="Wpisz nazwę katalogu:"
if exist %kat% echo Katalog %kat% już istnieje! & goto :END
MD %kat%
:END

```

Pora na wyjaśnienie, co i jak wykonuje program.

Pierwsza linia rozpoczyna się od dwukropka. Oznacza on, że wszystko to, co znajduje się na prawo od niego, powinno być traktowane jak komentarz. Drugi dwukropek ułatwia odnalezienie komentarza w długim programie. Po dwukropkach wpisana jest nazwa programu.

Druga linia również rozpoczyna się od dwóch dwukropków. Po niej zapisany jest opis programu.

Linia

```
@echo off
```

rozpoczyna się od znaku @. Zapobiega on wyświetleniu na początku polecenia pojedynczego wiersza z pliku wsadowego.

Polecenie `echo off` powoduje wyłączenie wyświetlania na ekranie poleceń wydawanych komputerowi przez program.

Linia

```
set /P kat="wpisz nazwę katalogu:"
```

zawiera polecenie `set`. Ma ono następującą składnię:

```
SET /P zmienna=[ciąg_monitu]
```

Przełącznik `/P` pozwala przypisać zmiennej ciąg znaków wprowadzony przez użytkownika. Przed odczytaniem ciągu znaków wyświetlany jest komunikat.



Wskazówka

Ciąg monitu w poleceniu może być pusty.

W analizowanym programie odczytanie powoduje wyświetlenie polecenia wpisania nazwy katalogu, a po wpisaniu tej nazwy — przypisanie zmiennej `kat`.

Polecenie

```
if exist %kat% echo Katalog %kat% już istnieje! & goto :END
```

powoduje sprawdzenie, czy istnieje już folder o takiej nazwie, jaka przypisana została zmiennej `kat`. Jeżeli folder istnieje, pojawia się komunikat i wykonywany jest skok do etykiety `:END`. Etykieta `:END` znajduje się na końcu programu. Skok do niej oznacza zakończenie wykonywania programu.

Jeżeli wprowadzona nazwa folderu jeszcze nie istnieje, wykonywana jest instrukcja

```
MD %kat%
```

Rezultatem jej wykonania jest założenie katalogu o nazwie przypisanej zmiennej `%kat%`.



Wskazówka

Zmiennej `kat` można przypisać nie tylko nazwę pojedynczego katalogu, ale i ścieżkę dostępu.

Po objaśnieniach przyszła pora na sprawdzenie działania programu.

1. Uruchom okno konsoli.
2. Przejdź do folderu, w którym znajduje się plik *kat.bat*.
3. Wpisz w oknie konsoli *kat*.
4. Naciśnij klawisz *Enter*.
5. Sprawdź, czy folder został utworzony (rysunek 3.20).

Rysunek 3.20.

Polecenie zostało wykonane

```

cmd.exe — skrót
F:\ftp>kat
Wpisz nazwe katalogu:Testowy
F:\ftp>tree
Zmienna PATH folderu
Numer seryjny woluminu: 28E3-AC73
F:
├── Testowy
F:\ftp>

```

Przykład 3.10.

Napisz program, który raz wywołany będzie zakładał foldery o nazwie podanej po wyświetleniu pytania. Program będzie działał aż do przerwania jego pracy przez użytkownika.

Polecenie jest wykonywane przez program:

```

:: kat1.bat
:: Program zakłada katalogi o nazwie wpisanej w odpowiedzi na pytanie.

@echo off

:START
set /P kat="Wpisz nazwe katalogu:"
if exist %kat% echo Katalog %kat% już istnieje! & goto :END
MD %kat%
:END
GOTO START

```

Program z ćwiczenia 3.9 był mało efektywny. Zawierał kilka linijek kodu. Pozwalał na wykonanie tego, do czego wystarczyło jedno polecenie. Czas poświęcony na jego napisanie nie był jednak zmarnowany. Pozwolił na przetestowanie programu. Po dodaniu pętli otrzymamy program, który realizuje zadanie podane na wstępie tego ćwiczenia.

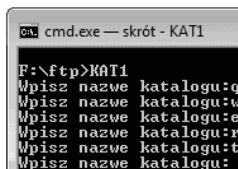
Polecam taką technikę pracy. Modyfikowanie programów przez dodawanie jednego elementu pozwala łatwo zlokalizować przyczynę błędu. Znajduje się ona w dodanym fragmencie lub dodany fragment wytworzył takie warunki działania programu, że ujawnił się błąd istniejący w już przetestowanej części.

Po objaśnieniach przyszła pora na sprawdzenie działania programu.

1. Uruchom okno konsoli.
2. Przejdź do folderu, w którym znajduje się plik *kat1.bat*.
3. Wpisz w oknie konsoli *kat1*.
4. Naciśnij klawisz *Enter*.
5. Wpisz nazwę katalogu, który chcesz utworzyć. Naciśnij klawisz *Enter*. Powtarzaj tę sekwencję czynności, aż utworzone zostaną wszystkie katalogi, które chcesz założyć (rysunek 3.21).
6. Aby zakończyć działanie programu, naciśnij klawisze *Ctrl+C*.

Rysunek 3.21.

Dzięki pętli program umożliwia założenie wielu katalogów po jednym ruchomieniu

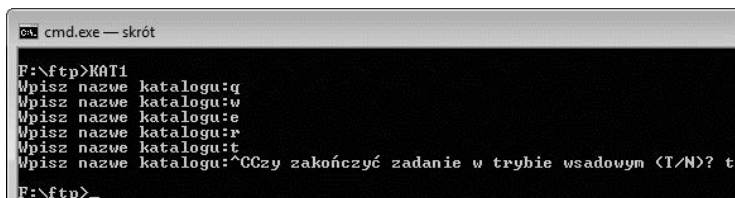


```
cmd.exe — skrót - KAT1
F:\ftp>KAT1
Wpisz nazwe katalogu:q
Wpisz nazwe katalogu:w
Wpisz nazwe katalogu:e
Wpisz nazwe katalogu:r
Wpisz nazwe katalogu:t
Wpisz nazwe katalogu:
```

7. Po wyświetleniu pytania o zakończenie pracy w trybie wsadowym naciśnij klawisz oznaczony literą *t*.
8. Działanie programu można zakończyć w dowolnej chwili (rysunek 3.22).

Rysunek 3.22.

Działanie programu można zakończyć, naciskając klawisze Ctrl+C



```
cmd.exe — skrót
F:\ftp>KAT1
Wpisz nazwe katalogu:q
Wpisz nazwe katalogu:w
Wpisz nazwe katalogu:e
Wpisz nazwe katalogu:r
Wpisz nazwe katalogu:t
Wpisz nazwe katalogu:^CCzy zakończyć zadanie w trybie wsadowym (Y/N)? t
F:\ftp>
```

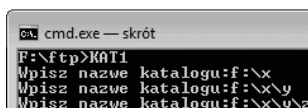
Przykład 3.11.

Sprawdź, czy w ćwiczeniu 3.10 jako nazwę folderu można wpisać pełną ścieżkę dostępu.

1. Uruchom okno konsoli.
2. Przejdź do folderu, w którym znajduje się plik *kat1.bat*.
3. Wpisz w oknie konsoli *kat1*.
4. Naciśnij klawisz *Enter*.
5. Wpisz nazwę katalogu, który chcesz utworzyć. Naciśnij klawisz *Enter*. Powtarzaj tę sekwencję czynności, aż utworzone zostaną wszystkie katalogi, które chcesz założyć (rysunek 3.23).

Rysunek 3.23.

Program pozwala na wskazanie pełnej ścieżki dostępu

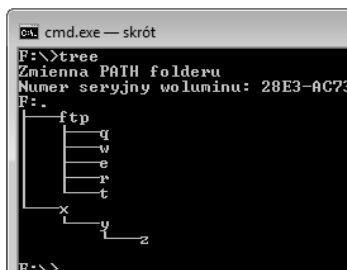


```
cmd.exe — skrót
F:\ftp>KAT1
Wpisz nazwe katalogu:f:\x
Wpisz nazwe katalogu:f:\x\y
Wpisz nazwe katalogu:f:\x\y\z
```

6. Sprawdź, jakie foldery zostały utworzone w wyniku wykonania przykładów 3.10 i 3.11 (rysunek 3.24).

Rysunek 3.24.

Foldery utworzone w wyniku wykonania przykładów 3.10 i 3.11



```
cmd.exe — skrót
F:\>tree
Zmiena PATH folderu
Numer seryjny woluminu: 28E3-AC73
F:\
├── ftp
│   ├── q
│   ├── w
│   ├── e
│   ├── r
│   └── t
├── x
│   ├── y
│   └── z
└──
```

mkdir

Polecenie `mkdir` umożliwia założenie katalogu.

Ma ono następującą składnię:

```
MKDIR [dysk:]ścieżka
```

gdzie:

dysk — litera napędu,

ścieżka — ścieżka dostępu do zakładanego katalogu.



Przy włączonych rozszerzeniach poleceń polecenie `MKDIR` tworzy wszystkie pośrednie katalogi w ścieżce.

Przykłady są takie same jak w przypadku polecenia `md`. Inna jest tylko nazwa polecenia.

3.2. Jak zmieniać nazwy katalogów?

Zmiana nazwy katalogu jest czynnością identyczną z przeniesieniem zawartości folderu do folderu o innej nazwie.

move

Polecenie `move` przenosi pliki oraz zmienia nazwy plików i katalogów.

Ma ono następującą składnię:

- ◆ przeniesienie jednego lub więcej plików:

```
MOVE [/Y | /-Y] [dysk:][ścieżka]nazwa_pliku1[...] cel
```

- ◆ zmiana nazwy katalogu:

```
MOVE [/Y | /-Y] [dysk:][ścieżka]katalog1 katalog2
```

gdzie:

`[dysk:][ścieżka]nazwa_pliku1` — ścieżka i nazwy plików do przeniesienia.

cel — nowa lokalizacja plików; może być oznaczona literą dysku z dwukropkiem, nazwą katalogu lub ich kombinacją.



Jeżeli przenoszony jest tylko jeden plik, w poleceniu można podać jego nową nazwę.

`[dysk:][ścieżka]katalog1` — nazwa katalogu, którą chcesz zmienić.

katalog2 — nowa nazwa katalogu.

/Y — wyłącza monitowanie o potwierdzenie zastąpienia istniejącego pliku docelowego.

/-Y — włącza monitowanie o potwierdzenie zastąpienia istniejącego pliku docelowego.

Przykład 3.12.

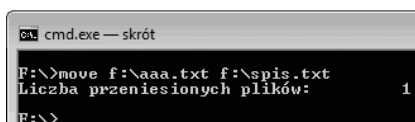
Zmień nazwę pliku *aaa.txt* znajdującego się na dysku *F* na *spis.txt*.

Aby wykonać polecenie:

1. Uruchom konsolę.
2. Wpisz z klawiatury `move f:\aaa.txt f:\spis.txt`.
3. Naciśnij klawisz *Enter*.
4. Nazwa pliku zostanie zmieniona (rysunek 3.25).

Rysunek 3.25.

*Okno konsoli
po zmianie nazwy pliku*

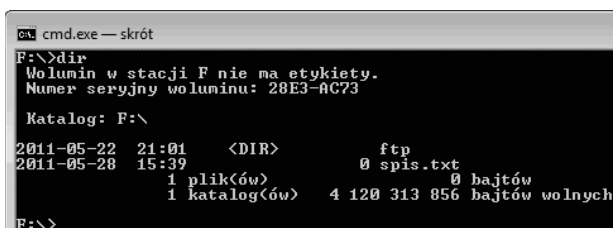


```
cmd.exe — skrót
F:\>move f:\aaa.txt f:\spis.txt
Liczba przeniesionych plików:      1.
F:\>
```

5. Sprawdź, jakie pliki znajdują się w partycji *F* (rysunek 3.26).

Rysunek 3.26.

*Do wyświetlania
plików najlepiej
nadaje się polecenie dir*



```
cmd.exe — skrót
F:\>dir
Wolumin w stacji F nie ma etykiety.
Numer seryjny woluminu: 28E3-AC73

Katalog: F:\

2011-05-22  21:01    <DIR>          ftp
2011-05-28  15:39                0 spis.txt          0 bajtów
                1 plik(ów)
                1 katalog(ów)  4 120 313 856 bajtów wolnych
F:\>_
```

Przykład 3.13.

Zmień nazwę znajdującego się na dysku *F* folderu *ftp* na *test*.

Aby wykonać polecenie:

1. Uruchom konsolę.
2. Wpisz z klawiatury `move f:\ftp f:\test`.
3. Naciśnij klawisz *Enter*.
4. Nazwa folderu zostanie zmieniona (rysunek 3.27).

Rysunek 3.27.

Okno konsoli
po zmianie
nazwy pliku

```
cmd.exe — skrót
F:\>move /Y f:\ftp f:\test
Liczba przeniesionych katalogów: 1.

F:\>tree
Zmienna PATH folderu
Numer seryjny woluminu: 28E3-AC73
F:.\_test
F:\>
```

Przykład 3.14.

Napisz program, który będzie przenosił pliki pomiędzy dwoma folderami. Nazwy folderów: źródłowego i docelowego będą parametrami wywołania programu.



Wskazówka

Zmienne oznaczane są w plikach wsadowych cyframi od 0 do 9.

Zapisz w pliku wsadowym *przenies.bat* program:

```
:: przenies.bat
:: Program przenosi pliki z katalogu źródłowego do docelowego.

@echo off
if %2X==X goto pomoc
move %1 %2
Echo Pliki %1 zostaną przeniesione do %2
goto koniec
Rem ---
: pomoc
Echo Program powinien być wywołany z dwoma parametrami
<pliki do przeniesienia><katalog docelowy>
Rem ---
: koniec
Echo on
```

Polecenie `@echo off` wyłącza echo. Od tej pory na ekranie nie będą wyświetlane wydawane polecenia.

Polecenie `if %2X==X goto pomoc` zawiera instrukcję warunkową. Sprawdza ona, czy pierwszy parametr i drugi parametr wywołania programu są identyczne. Jeśli tak — wykonywany jest skok do etykiety `:pomoc`.

Jeżeli pierwszy i drugi parametr wywołania programu są różne, wykonywana jest instrukcja `move %1 %2`. Zmiennej `%1` przypisana jest nazwa obiektu źródłowego. Zmiennej `%2` przypisana jest nazwa obiektu docelowego. Parametry te zostały pobrane z wywołania programu.

Wykonanie polecenia zapisanego w linii `Echo Pliki %1 zostaną przeniesione do %2` powoduje wyświetlenie komunikatu o przenoszeniu plików.

Wyświetlenie komunikatu jest ostatnią czynnością wykonywaną przez program w części związanej z przenoszeniem plików. Po wykonaniu kopiowania i wyświetleniu komunikatu wykonywana jest instrukcja `goto koniec`. Jej rezultatem jest skok do etykiety `koniec`.

Linia `Rem ---` zawiera komentarz. W tym przypadku trzy kreski oddzielają graficznie część programu z komunikatem o błędzie od reszty kodu.

Linie rozpoczynające się od znaku `:` zawierają etykiety. Są to wyróżnione miejsca w programie, do których wykonywane są skoki. W programie są dwie etykiety: `pomoc` i `koniec`.

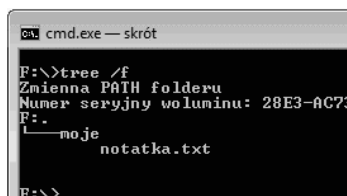
Po etykiecie `:` `pomoc` znajduje się tekst komunikatu: `Echo Program powinien być wywołany z dwoma parametrami <pliki do przeniesienia><katalog docelowy>`. Wyświetlany jest on tylko wtedy, gdy plik źródłowy i docelowy są identyczne.

Ostatnim poleceniem jest `Echo on`. Powoduje ono ponowne włączenie echa.

1. Utwórz folder o nazwie *moje*. W folderze *moje* umieść plik *notatka.txt* (rysunek 3.28).

Rysunek 3.28.

Dane testowe

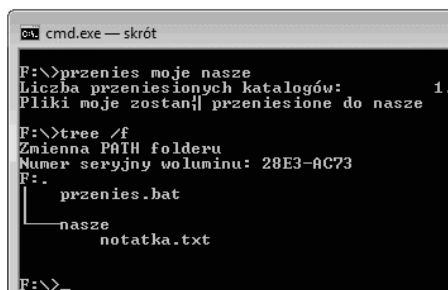


```
cmd.exe — skrót
F:\>tree /f
Zmienna PATH folderu
Numer seryjny woluminu: 28E3-AC73
F:.
├── moje
│   └── notatka.txt
F:\>
```

2. Umieść plik wsadowy z programem w katalogu głównym.
3. Wpisz w oknie konsoli polecenie `przenies moje nasze`.
4. Naciśnij klawisz `Enter`.
5. Sprawdź, czy nazwa folderu została zmieniona z *moje* na *nasze*, a plik *notatka.txt* został przeniesiony do nowego folderu (rysunek 3.29).

Rysunek 3.29.

Nazwa folderu została zmieniona. Zawartość na tym nie ucierpiała



```
cmd.exe — skrót
F:\>przenies moje nasze
Liczba przeniesionych katalogów: 1.
Pliki moje zostaną przeniesione do nasze
F:\>tree /f
Zmienna PATH folderu
Numer seryjny woluminu: 28E3-AC73
F:.
├── przenies.bat
└── nasze
    └── notatka.txt
F:\>
```

Przykład 3.15.

Napisz program, który będzie przenosił pliki z folderu źródłowego do folderów docelowych w zależności od rozszerzenia pliku.



Wskazówka

Zmienne oznaczane są w plikach wsadowych cyframi od 0 do 9.

Zapisz w pliku wsadowym *sort.bat* program:

```
:: sort.bat
:: Program sortuje pliki w zależności od rozszerzenia.

@echo off
move *.txt pliki_txt\
move *.xls pliki_xls\
move *.bmp pliki_bmp\
Echo on
```

Polecenie `@echo off` wyłącza echo. Od tej pory na ekranie nie będą wyświetlane wydawane polecenia.

Polecenie `move *.txt pliki_txt\` powoduje przeniesienie pliku o rozszerzeniu **.txt* do folderu *pliki_txt*. Jeśli folder *pliki_txt* nie istnieje, zostanie utworzony.

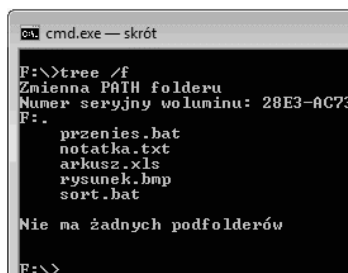
W analogiczny sposób działają polecenia `move *.xls pliki_xls\` i `move *.bmp pliki_bmp\`.

Ostatnim poleceniem jest `Echo on`. Powoduje ono ponowne włączenie echa.

1. Utwórz pliki o rozszerzeniach **.txt*, **.xls*, **.bmp*. W tym samym folderze umieść plik *sort.bat* (rysunek 3.30).

Rysunek 3.30.

Dane testowe



```
cmd.exe — skrót
F:\>tree /f
Zmienna PATH folderu
Numer seryjny woluminu: 28E3-AC73
F:.
  przenies.bat
  notatka.txt
  arkusz.xls
  rysunek.bmp
  sort.bat

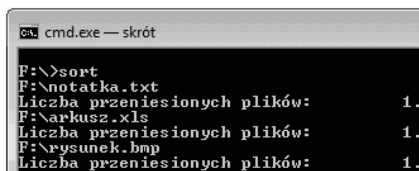
Nie ma żadnych podfolderów
F:\>
```

2. Wpisz w oknie konsoli polecenie `sort`.
3. Naciśnij klawisz *Enter*.
4. Polecenie zostanie wykonane (rysunek 3.31).

Rysunek 3.31.

Informacja

o przeniesieniu plików

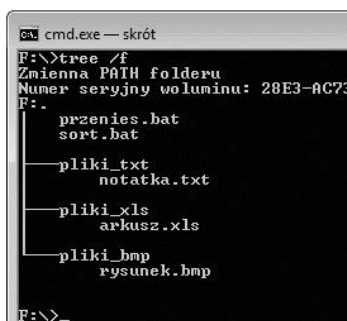


```
cmd.exe — skrót
F:\>sort
F:\notatka.txt
Liczba przeniesionych plików: 1.
F:\arkusz.xls
Liczba przeniesionych plików: 1.
F:\rysunek.bmp
Liczba przeniesionych plików: 1.
```

5. Sprawdź, gdzie zostały przeniesione pliki (rysunek 3.32).

Rysunek 3.32.

*Pliki trafiły
do odpowiednich
folderów*



```
cmd.exe — skrót
F:\>tree /f
Zmienna PATH folderu
Numer seryjny woluminu: 28E3-AC73
F:.
├── przenies.bat
│   └── sort.bat
├── pliki_txt
│   └── notatka.txt
├── pliki_xls
│   └── arkusz.xls
└── pliki_bmp
    └── rysunek.bmp
F:\>
```

3.3. Jak usuwać katalogi?

Jeżeli piszesz program, który do działania wymaga utworzenia folderów roboczych, przed zakończeniem jego pracy należy foldery usuwać. Istnieje instrukcja, która pozwala na usunięcie całego drzewa katalogów.

rd

Polecenie `rd` umożliwia usuwanie katalogów.

Ma ono następującą składnię:

```
RD [/S] [/Q] [dysk:]ścieżka
```

gdzie:

`/S` — usuwa określony katalog wraz ze wszystkimi znajdującymi się w nim katalogami i plikami.

`/Q` — nie jest wymagane potwierdzanie usuwania plików i folderów, jeśli została użyta opcja `/S`. Jest to tzw. tryb cichy.



Wskazówka

Identyczne działanie i składnię ma polecenie `rmdir`

Przykład 3.16.

Napisz program, który sprawdza istnienie katalogu i w zależności od tego, czy istnieje, usuwa go lub tworzy.

Zapisz w pliku wsadowym *kat2.bat* program:

```
:: kat2.bat
:: Program sprawdza istnienie katalogu i w zależności od tego,
:: czy istnieje, usuwa go bądź tworzy.
```

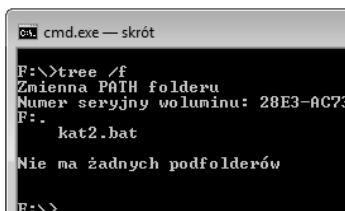
```
@echo off
if exist \tmp\nul (rd \tmp) else (md \tmp)
Echo on
```

Zasadniczą częścią programu jest linia z poleceniem `if`. Nie można go używać do bezpośredniego sprawdzenia istnienia katalogu. W każdym katalogu istnieje urządzenie zerowe `nul`. Można więc sprawdzić, czy istnieje urządzenie zerowe. Stąd będzie wiadomo, czy istnieje katalog.

1. Umieść plik *kat2.bat* w katalogu głównym napędu, w którym chcesz zarządzać pojawianiem się i znikaniem pliku *tmp*.
2. Sprawdź zawartość katalogu głównego (rysunek 3.33).

Rysunek 3.33.

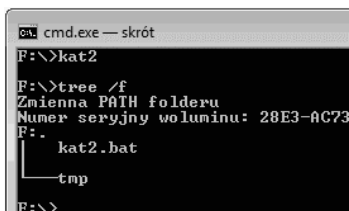
*W katalogu głównym
nie ma folderów*



3. Wpisz z klawiatury *kat2*.
4. Naciśnij klawisz *Enter*.
5. Sprawdź zawartość katalogu głównego (rysunek 3.34).

Rysunek 3.34.

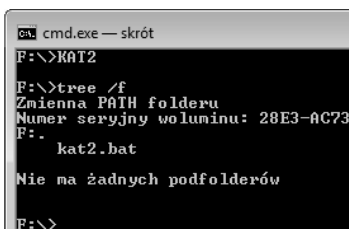
*Utworzony został
katalog tmp*



6. Wpisz z klawiatury *kat2*.
7. Naciśnij klawisz *Enter*.
8. Sprawdź zawartość katalogu głównego (rysunek 3.35).

Rysunek 3.35.

*Katalog tmp został
usunięty*



Przykład 3.17.

Napisz program, który będzie usuwał z napędu wszystkie pliki i foldery wraz z zawartością. Operacja ma nie wymagać potwierdzenia. Litera napędu będzie parametrem uruchomienia programu.

1. Zapisz w pliku wsadowym *usun.bat* program:

```
:: usun.bat
:: Program usuwa wszystkie dane napędu, który jest parametrem
:: wywołania programu.
@echo off
cls
Echo Program usunie wszystkie dane z napędu %1
Pause
rd /s /q %1:\
Echo on
```

Pierwsze dwie linie programu zawierają jego nazwę i opis.

Polecenie

```
@echo off
```

powoduje wyłączenie wyświetlania na ekranie linii programu. Włączenie wyświetlania wykonywanych poleceń jest realizowane przez ostatnią linię programu.

Polecenie

```
cls
```

ma za zadanie usunięcie z okna konsoli wszystkich napisów.

Polecenie

```
Echo Program usunie wszystkie dane z napędu %1
```

powoduje wyświetlenie na ekranie komunikatu, który rozpoczyna się od słów *Program usunie....*

Polecenie

```
Pause
```

powoduje wyświetlenie na ekranie polecenia Aby kontynuować, naciśnij dowolny klawisz.... Nośnik nie jest jeszcze czyszczony. Usuwanie jego zawartości rozpocznie się po naciśnięciu klawisza.

Polecenie

```
rd /s /q %1:\
```

powoduje usunięcie wszystkich plików i katalogów znajdujących się w napędzie, którego oznaczenie literowe przypisane jest zmiennej %1. Przełączniki powodują, że usuwanie będzie wykonywane wraz z folderami (/s) oraz nie będzie wymagało potwierdzenia (/q).

2. Umieść plik *usun.bat* w katalogu głównym napędu.

3. Wyświetl okno konsoli.
4. Przejdź do napędu, w którym umieszczony jest plik *usun.bat*.
5. Wyświetl zawartość napędu.
6. Wpisz z klawiatury `usun f` (rysunek 3.36).

Rysunek 3.36.

W napędzie F znajduje się plik przeznaczony do usuwania plików i folderów

```

cmd.exe — skrót
F:\>tree /f
Zmienna PATH folderu
Numer seryjny woluminu: 28E3-AC73
F:\.
  usun.bat
Nie ma żadnych podfolderów
F:\>usun f
  
```

7. Naciśnij klawisz *Enter*.
8. Program rozpocznie działanie. Jest to ostatnia szansa na przerwanie jego pracy bez usuwania plików i folderów. Aby wykonać program, należy nacisnąć dowolny klawisz (rysunek 3.37).

```

cmd.exe — skrót - usun f
F:\>usun f
Program usunie wszystkie dane z napędu f
Aby kontynuować, naciśnij dowolny klawisz . . .
F:\>
  
```

Rysunek 3.37. *Przerwać działanie programu można, naciskając klawisze Ctrl+C*

9. Program zostanie wykonany (rysunek 3.38).

Rysunek 3.38.

Nie udało się odnaleźć pliku wsadowego. Cóż takiego się stało?

```

cmd.exe — skrót
F:\>usun f
Program usunie wszystkie dane z napędu f
Aby kontynuować, naciśnij dowolny klawisz . . .
Znalezienie pliku wsadowego nie było możliwe.
F:\>
  
```

10. Wyświetl zawartość napędu (rysunek 3.39).

Rysunek 3.39.

Zagadka rozwiązana! Program sprzątał tak dokładnie, że... posprzątał również siebie

```

cmd.exe — skrót
F:\>tree /f
Zmienna PATH folderu
Numer seryjny woluminu: 28E3-AC73
F:\.
Nie ma żadnych podfolderów
F:\>
  
```



Wskazówka

Program nieodwracalnie oczyszcza nośnik umieszczony w napędzie. Należy zwrócić uwagę, by przez przypadek nie usunąć ewentualnych ważnych danych.

Przykład 3.18.

Napisz program, który z dyskielek kolejno umieszczanych w napędzie *A* będzie usuwał wszystkie pliki i foldery wraz z zawartością. Operacja ma nie wymagać potwierdzenia.

W pliku wsadowym *usuwacz.bat* zapisz program:

```
:: usuwacz.bat
:: Program usuwa wszystkie dane z dyskietki umieszczonej w napędzie A
@echo off
: start
cls
Echo Włóż dyskietkę do napędu.
pause
cls
rd /s /q a:\
Echo Dyskietka została wyczyszczona.
pause
goto start
```

Program *usuwacz.bat* jest podobny do *usun.bat*. Zasadnicza różnica polega na tym, że *usuwacz.bat* działa w pętli. W czwartej linii widoczna jest etykieta `: start`. Do niej wykonywany jest skok z ostatniej linii programu. Za wykonanie skoku odpowiedzialna jest instrukcja `goto start`.

Po wyczyszczeniu jednej dyskietki program jest gotowy do wyczyszczenia kolejnej. Nie trzeba go ponownie uruchamiać. Aby przerwać działanie programu *usuwacz.bat*, należy nacisnąć klawisze *Ctrl+C*.

Skorowidz

-, 89, 98
\$, 84
%, 41
%~\$PATH:1, 133
%~1, 133
%~a1, 133
%~d1, 133
%~dp\$PATH:1, 134
%~dp1, 133
%~fl, 133
%~ftzal, 134
%~n1, 133
%~nx1, 133
%~p1, 133
%~s1, 133
%~t1, 133
%~x1, 133
%~z1, 133
%ALLUSERSPROFILE%, 273
%APPDATA%, 273
%CD%, 45, 273
%CMDCMDLINE%, 45, 273
%CMDEXTVERSION%, 45, 273
%COMPUTERNAME%, 273
%COMSPEC%, 273
%DATE%, 45, 273
%ERRORLEVEL%, 45, 273
%HOMEDRIVE%, 273
%HOMEPATH%, 273
%HOMESHARE%, 273
%LOGONSERVER%, 273
%NUMBER_OF_
 ↳PROCESSORS%, 273
%OS%, 273
%PATH%, 274
%PATHEXT%, 274
%PROCESSOR_
 ↳ARCHITECTURE%, 274

%PROCESSOR_IDENTFIER%, 274
%PROCESSOR_LEVEL%, 274
%PROCESSOR_
 ↳REVISION%, 274
%PROMPT%, 274
%RANDOM%, 45, 274
%SYSTEMDRIVE%, 274
%SYSTEMROOT%, 274
%TEMP%, 274
%TIME%, 45, 274
%TMP%, 274
%USERDOMAIN%, 274
%USERNAME%, 274
%USERPROFILE%, 274
%WINDIR%, 274
&, 22, 272
&&, 22, 272
(), 272
(CI), 103
(I), 103
(IO), 103
(NP), 103
(OI), 103
*, 84
*.bat, 15
., 84
:, 53
;, 272
@, 46
[^klasa], 84
[klasa], 84
[x-y], 84
^, 84
||, 22, 272
+, 98
<, 73
<xyz, 84

>, 64, 73
>>, 73
2>, 73
2>>, 73
437, 175
850, 175
852, 175

A

A, 89, 98
Access Control List, 100
ACL, 100
AD, 103
add server, 265
adres
 IP, 226
 MAC, 223
aktualizacja bazy danych kont
 użytkowników, 231
alias, 241
ANSI, 22
aplikacja, 14
argumenty wywołania, 41
AS, 103
assoc, 96
at, 117
atrybut
 pliku, 98
 zanegowanie, 89
attrib, 98
automatyczne
 kopiowanie plików, 120
 uruchamianie przeglądarki,
 122
AutoRun, 23
AUX, 73

B

badanie zadań, 122
 batch, 15
 bez propagowania dziedziczenia (ACL), 103
 bieżące
 katalog, 59
 połączenie TCP/IP, 262
 blokady plików, 236
 brak dostępu (ACL), 102
 brama domyślna, 226
 break, 138

C

call, 132
 cd, 59
 chcp, 175
 chdir, 63
 chkdsk, 206
 chknfs, 208
 choinka, 183
 cipher, 214
 cls, 142
 cmd, 20, 22
 color, 173
 COM, 73
 comp, 106
 compact, 210
 COMPUTERNAME, 194
 CON, 73
 convert, 212
 copy, 109
 cscript, 159
 czas systemowy, 176

D

D, 103
 DACL, 101
 data, 177
 date, 177
 DC, 103
 DE, 103
 defrag, 205
 defragmentacja woluminu, 205
 dekompresja plików, 116
 del, 88
 delete server, 266
 DHCP, 226, 265
 dir, 36, 92, 94
 Disk Operating System, 13

diskcomp, 114
 diskcopy, 115
 diskperf, 217
 DNS, 226
 dodanie
 danych/ podkatalogu (ACL), 103
 komputerów do bazy danych domeny, 234
 serwera DHCP do listy autoryzowanych serwerów Active Directory, 265
 dokańczanie nazw plików i katalogów, 32
 Domain Name System, 226
 DOS, 13
 doskey, 178
 dostęp
 do modyfikowania (ACL), 102
 do odczytu i wykonywania (ACL), 102
 do usuwania (ACL), 103
 tylko do odczytu (ACL), 102
 tylko do zapisu (ACL), 103
 zabezpieczeń systemu (ACL), 103
 driverquery, 196
 drukowanie w tle, 87
 drzewo
 folderów, 64
 katalogów (kopiowanie), 66
 plików, 64
 dump, 264, 266
 Dynamic Host Configuration Protocol, 226
 dysk
 systemowy, 194
 wirtualny, 195
 dziedziczenie
 kontenera (ACL), 103
 obiektu (ACL), 103
 dziennik statystyk usługi, 248

E

echo, 23, 183
 endlocal, 154
 EQU, 149
 erase, 92
 errorlevel, 134
 ERRORLEVEL, 147

etykieta, 53
 woluminu
 odczyt, 203
 zmiana, 203
 exit, 161
 expand, 116

F

F, 102
 F1 (klawisz funkcyjny), 34
 F2 (klawisz funkcyjny), 34
 F3 (klawisz funkcyjny), 34
 F4 (klawisz funkcyjny), 34
 F5 (klawisz funkcyjny), 34
 F6 (klawisz funkcyjny), 34
 F7 (klawisz funkcyjny), 34
 F8 (klawisz funkcyjny), 34
 F9 (klawisz funkcyjny), 34
 FAT, 200
 FAT32, 200
 fc, 107
 file allocation table, 199
 find, 77, 277, 280
 findstr, 81
 flushdns, 227
 folder
 struktura, 63
 zakładanie, 35
 for, 143
 format, 199
 formatowanie nośników, 199
 fsutil, 218
 ftype, 97

G

GA, 103
 GE, 103
 GEQ, 149
 getmac, 223
 goto, 146
 gotowy do archiwizacji (atrybut), 89
 GR, 103
 grupy
 globalne w domenach, 237
 lokalne, 239
 GTR, 149
 GW, 103

H

H, 89, 98
hardware, 13
Harmonogram zadań (usługa), 118
help, 185
HOMEDRIVE, 194
HOMEPATH, 194
HOMESHARE, 194
host skryptów
 systemu Windows, 159

I

I, 89, 98
icacls, 100
identyfikator SID, 102, 128
if, 147
informacja o
 komputerze, 261
 poleceniu, 185
 połączeniu sieciowym, 231
 serwerze, 235
 stacji roboczej, 235
Insert, 179
instalacja woluminu, 201
instrukcja warunkowa, 147
ipconfig, 226
IPX/SPX, 17

J

JScript, 159

K

katalog
 bieżący, 59
 nadpisanie, 37
 struktura, 63
 zakładanie, 35
 zawierający system, 194
 zmiana nazwy, 88
 zmiana, 59
katalogi przeszukiwane przy
 uruchamianiu programów
 z linii komend, 194
katalogowanie zawartości
 nośnika, 93
kod błędu, 147
kolejki wydruku, 242
kolor
 pierwszego planu, 23, 173
 tła, 23, 173

komentarz, 53
kompresja na partycjach NTFS,
 210
konfiguracja
 bazy danych IAS, 264
 serwera DHCP, 266
 sieciowa, 229
 uruchomionego
 komputera, 263
 wczytanie z pliku, 230
 zapisanie do pliku, 229
konsola, 20
 kolory, 24
 zakończenie pracy, 26
kontrola odczytu (ACL), 103
kontynuacja usługi wstrzymanej,
 236
konwersja woluminu FAT
 na NTFS, 212
kończenie zadań, 122
kopiowanie
 automatyczne, 120
 drzewa katalogów, 66
 dyskietki, 115
 plików, 66, 109
Kosz, 88
kwalifikator, 271

L

L, 89
label, 203
LEQ, 149
lista
 aplikacji i skojarzonych
 zadań (procesów), 165
 domen udostępnionych przez
 komputer, 255
 komputerów udostępnionych
 przez komputer, 255
 kontroli dostępu, 100
 otwartych plików, 164
 plików, 92
 podkatalogów, 92
 poleczeń sieciowych, 239
 rozszerzeń akceptowanych
 jako programy, 194
 zadań
 uruchomionych, 129, 166
 zaplanowanych do
 uruchomienia, 121
 zainstalowanych
 sterowników, 196
 zasobów udostępnionych
 przez komputer, 255

litera dysku z katalogiem
 domowym, 194
local, 164
LPT, 73
LSS, 149

M

M, 102
MA, 103
maksymalny dozwolony (ACL),
 103
maska podsieci, 226
md, 35
Microsoft Disk Operating
 System, 13
minimalna długość haseł
 do kont użytkowników, 232
mkdir, 50
mode, 187
modyfikacja
 kont użytkowników, 252
 zadań, 122
modyfikator, 271
more, 80, 150, 277
mountvol, 201
move, 50, 88
MS-DOS, 13
msg, 260

N

N, 102
nazwa
 bieżącego
 katalogu, 59
 użytkownika, 194
 domeny, z jakiej pochodzi
 bieżący użytkownik, 194
 komputera, 194
 nazwy wszystkich otwartych
 plików udostępnionych
 na serwerze, 236
nbtstat, 262
NEQ, 149
net, 231
 computer, 234
 config, 235
 continue, 236
 file, 236
 group, 237
 help, 239
 helpmsg, 239
 localgroup, 239
 name, 241

net

pause, 236, 241
 print, 242
 send, 243
 session, 244
 share, 246
 start, 247
 statistics, 248
 stop, 249
 time, 249
 use, 250
 user, 252
 view, 255

NetBEUI, 17

netsh, 229, 263, 265

netstat, 258

niezindeksowany (atrybut), 89

NUL, 73

numer

seryjny, 203
 strony kodowej konsoli, 189

O

odczyt

atrybutów (ACL), 103
 rozszerzonych (ACL), 103
 danych/wyświetlanie
 katalogu (ACL), 103
 rodzajowy (ACL), 103

odinstalowanie woluminu, 218

Odmowa dostępu (komunikat),
 121

odzyskiwanie danych
 z nieuszkodzonych
 sektorów, 210

openfiles, 162

opóźnione rozwijanie
 zmiennych

systemowych, 194
 środowiskowych, 23

otwarte pliki, 162, 163

Overstrike, 179

P

parametry czasowe połączenia
 sieciowego, 257

path, 32, 172

Path, 194

PATH, 193

PATHEXT, 194

pause, 136

pełny dostęp (ACL), 102

pętla, 143

ping, 257

pisanie makr, 178

plik

archiwizacji, 98

atrybuty, 89

kopiowanie, 66

niezindeksowanej

zawartości, 98

plik wsadowy, 11, 20, 25

uruchamianie, 12

przenoszenie, 88

systemowy, 98

tekstowy (zawartość), 74

tylko do odczytu, 98

ukryty, 98

wymiany, 16

zmiana nazwy, 88

podział czasu, 16

polecenia

uruchamianie w trybie
 tekstowym, 19

wpisywane z klawiatury,
 178

połączenie

kilku poleceń, 22

komputera z
 udostępnionym zasobem,
 250

serwerem, 244

sieciowe (parametry

czasowe), 257

pomiar wydajności dysków, 217

popd, 153

poprawność zapisywania

plików, 204

porównanie

dyskietek, 114

plików, 107

zawartości plików, 106

port szeregowy, 187

poufne listy kontroli dostęp, 101

praca

w trybie konsoli, 20

z podziałem czasu, 16

print, 87

PRN, 73

program

użytkowy, 14

wsadowy, 15

prompt, 170

protokół sieciowy, 17

przejsięcie do katalogu

zachowanego przez polecenie
 PUSHD, 153

przetwarzanie wsadowe, 15

przyczyny wystąpienia błędu,
 239

przywoływanie poleceń
 wpisywanych z klawiatury,
 178

punkt ponownej analizy
 (atrybut), 89

pushd, 153

Q

query, 163

R

R, 89, 98, 102

RA, 103

RC, 103

rd, 55

RD, 103

REA, 103

readresowanie drukowania, 188

recover, 210

Rem, 53

ren, 112

rename, 113

replace, 113

rozmiar buforu ekranu dla
 wiersza polecenia, 189

rozpakowanie plików
 instalacyjnych, 116

rozszerzanie woluminu, 218

rozszerzenia poleceń, 23, 28

rozszerzenia włączone, 89

rozszerzenie nazwy pliku, 96

rozszerzone sprawdzanie
 naciśnięcia kombinacji
 klawiszy Ctrl+C, 138

RX, 102

S

S, 89, 98, 103

SACL, 101

schtasks, 122

Serwer (usługa), 235

sesja terminalowa, 261

set, 191

set config, 264

setlocal, 154
shift, 157
show config, 264
show version, 265
shutdown, 167
SID, 102, 128
Simultaneous Pheripheral
 Operation On-Line, 16
skojarzenia rozszerzeń plików,
 96, 97
skoku do etykiety, 146
skrót (tworzenie), 10
skrypty WSH, 159
software, 14
sort, 85, 277, 281
sortowanie
 danych, 281
 wierszy w pliku, 85
spooling, 16
sprawdzanie
 dysku, 206
 podczas rozruchu, 208
 parametrów czasowych
 połączenia sieciowego, 257
 warunku, 147
sprzęt komputerowy, 13
Stacja robocza (usługa), 235
stan
 strony kodowej konsoli, 189
 urządzeń zainstalowanych
 w systemie, 188
standardowe urządzenie we/wy,
 73
start, 138
statystyka
 połączenia TCP/IP, 258
 protokołu TCP/IP, 262
sterowniki, 196
strona kodowa, 175
struktura katalogów, 63
subst, 195
swap file, 16
swapped
 in, 17
 out, 17
swapping, 16
synchronizacja zegara
 komputera z zegarem innego
 komputera lub domeny, 249
synchronizowanie (ACL), 103
system
 komputerowy, 13
 operacyjny, 14
 z podziałem czasu, 16

SystemDrive, 194
systeminfo, 261
systemowe listy kontroli
 dostępu, 101
systemowy (atrybut), 89
SystemRoot, 194
szybkość powtarzania
 klawiatury, 189
szyfrowanie na partycjach
 NTFS, 214

Ś

ścieżka
 bezwzględna, 30
 relatywna, 30
ścieżka do katalogu domowego,
 194
ścieżka do profilu bieżącego
 użytkownika, 194
ścieżka wyszukiwania dla plików
 wykonywalnych, 172

T

tablica alokacji plików FAT,
 199
taskkill, 166
tasklist, 165
TCP/IP, 17, 226
 konfiguracja statyczna, 268
 resetowanie ustawień, 267
 statystyka połączenia, 258
time, 176
time-sharing system, 16
title, 141
tree, 36, 63
tryb konsoli, 20
tryb tekstowy
 uruchamianie poleceń, 19
tryb zgłoszenia konsoli, 170
tscon, 261
tworzenie zadań, 122
tylko do odczytu (atrybut), 89
tylko dziedziczenie (ACL), 103
type, 74
tytuł okna konsoli, 141

U

ukryty (atrybut), 89
ukrywanie komunikatów, 183
UNC do katalogu domowego,
 194

Unicode, 23
uprawnienie odziedziczone
 z kontenera nadrzędnego
 (ACL), 103
uruchamianie programów
 o określonej godzinie, 117
uruchamianie skryptów za
 pomocą hosta skryptów, 159
uruchamianie usług, 247
uruchamianie zadań, 122
urządzenia wejścia-wyjścia, 16
urządzenia zewnętrzne, 187
urządzenie puste, 73
USERDOMAIN, 194
USERNAME, 194
USERPROFILE, 194
Usługa nie została uruchomiona
 (komunikat), 121
ustawienia usługi
 Serwer, 235
 Stacja robocza, 235
usunięcie serwera DHCP z listy
 autoryzowanych serwerów
 Active Directory, 266
usuwanie
 ACL, 103
 drzewa folderów, 39
 elementu podrzędnego
 (ACL), 103
 katalogów, 55
 komputerów z bazy danych
 domeny, 234
 plików, 88, 92
 z wyspecyfikowanego
 napędu lub folderu, 90
 wszystkich plików, 90
 zadań, 122
użytkownik, 14

V

VBScript, 159
ver, 169
verify, 204
vol, 203

W

W, 103
WA, 103
WD, 103
WDAC, 103
WEA, 103

wersja
 bazy danych IAS, 265
 systemu operacyjnego Windows, 169
 weryfikacja poprawności zapisywania plików, 204
 WO, 103
 wolumin, 201
 wsad, 15
 Wscript.exe, 159
 WSH, 159
 wstrzymanie usługi, 241
 wszelki dostęp rodzajowy (ACL), 103
 wybór strony kodowej, 188
 wydajność dysków, 217
 wyczyszczenie okna wiersza poleceń, 142
 wykonywanie
 rodzajowe (ACL), 103
 zadania w pętli, 143
 wyrażenia regularne, 84
 wysyłanie
 komunikatu do użytkownika, 260
 wiadomości, 243
 wyszukanie ciągu znaków, 280
 w pliku, 77, 81
 wyświetlania
 adresów MAC, 223
 komunikatów, 183
 menu, 134
 oddzielnego okna konsoli, 138
 zawartości pliku ekran po ekranie, 150
 wywołanie programu wsadowego, 132

X

x, 84
 X, 103
 xcopy, 66
 xyz, 84

Z

zachowanie bieżącego katalogu dla polecenia POPD, 153
 zadania, 122
 usuwanie, 129
 zakładanie katalogu, 35, 50
 zakończenie procesów, 166
 zamiana plików, 113
 zamykanie i ponowne uruchamianie systemu, 167
 zanegowanie atrybutu, 89
 zapamiętywanie poleceń wpisywanych z klawiatury, 178
 zapis
 atrybutów (ACL), 103
 rozszerzonych (ACL), 103
 DAC (ACL), 103
 danych/dodawanie pliku (ACL), 103
 rodzajowy (ACL), 103
 właściciela (ACL), 103
 zarządzanie
 punktami ponownej analizy, 218
 udostępnionymi zasobami, 246
 zatrzymanie usługi, 249
 zawartość pliku tekstowego, 74
 zawieszenie przetwarzania pliku wsadowego, 136

złącze
 równoległe, 73
 szeregowo, 73
 zmiana
 katalogu, 59
 konfiguracji sieciowej, 229
 nazwy
 katalogu, 50
 pliku, 50, 112, 113
 położenia wymiennych parametrów w pliku wsadowym, 157
 zmienne
 systemowe, 30
 środowiskowe, 30, 193, 194, 273
 dla plików wsadowych, 154
 programu cmd.exe, 191
 rozwijanie opóźnione, 23
 użytkownika, 30
 znaki dokończania nazw plików i katalogów, 23

PROGRAM PARTNERSKI

GRUPY WYDAWNICZEJ HELION



1. ZAREJESTRUJ SIĘ
2. PREZENTUJ KSIĄŻKI
3. ZBIERAJ PROWIZJĘ

Zmień swoją stronę WWW
w działający bankomat!

Dowiedz się więcej i dołącz już dzisiaj!

<http://program-partnerski.helion.pl>

GRUPA WYDAWNICZA

 **Helion SA**



Odkryj moc poleceń tekstowych systemu Windows 7!

Każdy nowoczesny system operacyjny ma charakter graficzny, jednak w każdym można korzystać również z linii poleceń. Za jej pomocą da się przeprowadzić wszystkie operacje dostępne z poziomu interfejsu, a także wiele innych. Tryb tekstowy nie jest zbyt popularny wśród przeciętnych użytkowników Windowsa, ponieważ wymaga znajomości odpowiednich komend, sposobów ich stosowania i parametrów sterujących, a ponadto pewnej wiedzy na temat działania samego systemu operacyjnego. Jest też zwykle uważany za trudniejszy. Jeśli jednak chcesz korzystać z pełni możliwości środowiska i poświęcać mniej czasu na pracę dzięki automatyzacji wykonywania często powtarzanych operacji, nic nie pomoże Ci bardziej niż dogłębna znajomość trybu tekstowego oraz umiejętność pisania programów wsadowych.

Pierwszy krok na drodze do zdobycia odpowiedniej wiedzy ułatwi Ci książka „Windows 7. Komendy i polecenia. Praktyczne przykłady”. Niezależnie od tego, czy jesteś początkującym użytkownikiem Windowsa, czy też korzystasz z kolejnych wersji tego systemu już od wielu lat, podręcznik ten otworzy przed Tobą zupełnie nowe możliwości, oferując Ci dostęp do potęgi kryjącej się w poleceniach trybu tekstowego. Poznasz sposoby zarządzania systemem plików, zadaniami, środowiskiem i połączeniami sieciowymi. Nauczysz się kontrolować podstawowe funkcje systemu operacyjnego i praktycznie opanujesz najważniejsze komendy tekstowe. Dowiesz się także, jak tworzyć i wykorzystywać pliki wsadowe, pozwalające z łatwością wykonywać całe sekwencje poleceń i automatyzować pracę. Oszczędź czas i energię!

- Podstawowe informacje na temat systemu operacyjnego
- Korzystanie z trybu tekstowego i dostosowywanie go do swoich potrzeb
- Przeprowadzanie operacji na plikach i katalogach
- Zarządzanie zadaniami i tworzenie plików wsadowych
- Kontrolowanie środowiska systemowego
- Zarządzanie systemem plików i operacje na napędach
- Konfigurowanie i testowanie połączeń sieciowych

Przekonaj się, co możesz osiągnąć za pomocą trybu tekstowego Windows 7!

helion.pl
księgarnia
internetowa

Nr katalogowy: 6306



Księgarnia internetowa
<http://helion.pl>



Zamówienia telefoniczne:

0 801 339900



0 601 339900



Helion

Sprawdź najnowsze promocje:
• <http://helion.pl/promocje>
Książki najchętniej czytane:
• <http://helion.pl/bestsellery>
Zamów informacje o nowościach:
• <http://helion.pl/nowosci>

Helion SA
ul. Kościuszki 1c, 44-100 Gliwice
tel.: 32 230 98 63
e-mail: helion@helion.pl
<http://helion.pl>

Cena 39,00 zł

ISBN 978-83-246-3537-5



9 788324 635375

Informatyka w najlepszym wydaniu