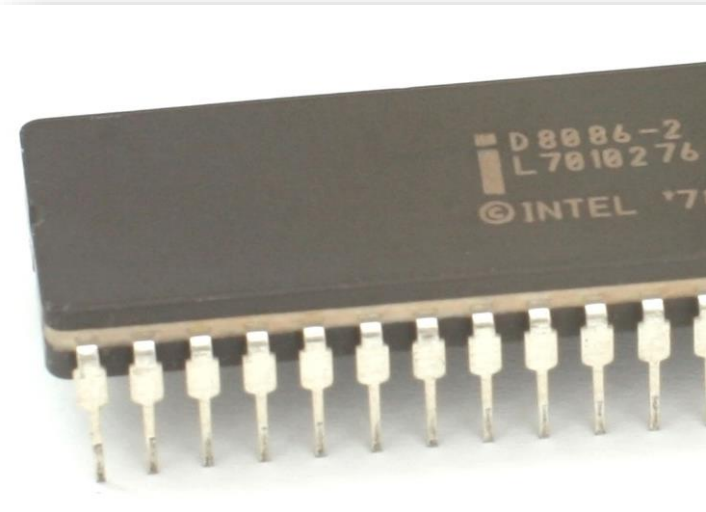


# Systemy operacyjne

# MS-DOS



```
Wang Professional Computer - BIOS 1.65  
Copyright, Wang Laboratories, Inc., 1982  
,83,84,85  
All rights reserved
```

```
MS-DOS v2.01a  
Copyright 1981,82,83 Microsoft Corp.  
AECLOCK v1.0.0 installed.
```

```
Command v. 2.10a  
Current date is Wed 1/18/2012  
Enter new date:  
Current time is 0:09:10.83  
Enter new time:
```

```
A:\loadpsf italic.psf
```

```
A:\ver
```

```
MS-DOS Version 2.01
```

```
A:
```

# Komputer, jako sprzęt

**nie jest** urządzeniem zdolnym wykonywać żądane **przez użytkownika zadania** - potrafi jedynie przetestować stan urządzeń wchodzących w jego skład oraz uruchomić system operacyjny.



# System operacyjny



to program kontrolujący pracę komputera:

- umożliwia uruchamianie programów użytkownika nadzorując ich wykonywanie i udostępniając im zasoby komputera (pamięć, czas procesora, drukarki, ...),
- umożliwia łatwe zapisywanie danych na dyskach, CD-ach
- komunikację z innymi komputerami w sieci,
- ....



# Plik

**to podstawowa jednostka manipulacji danymi  
poprzez system operacyjny** („w komputerze”).

jest to ciąg bitów (bajtów) danych, opatrzony nazwą i atrybutami.

Dane są zapisane w postaci **kodu binarnego**, zależnie od zawartości pliku. Nazwa z reguły składa się z głównej części nazwy i tzw. rozszerzenia (oddzielonego kropką).

**nazwa\_03.ext**

# Plik

System operacyjny składa się z plików. Wszystkie składniki systemu (także rejestry i ustawienia) przechowywane są w plikach (w pamięci masowej).

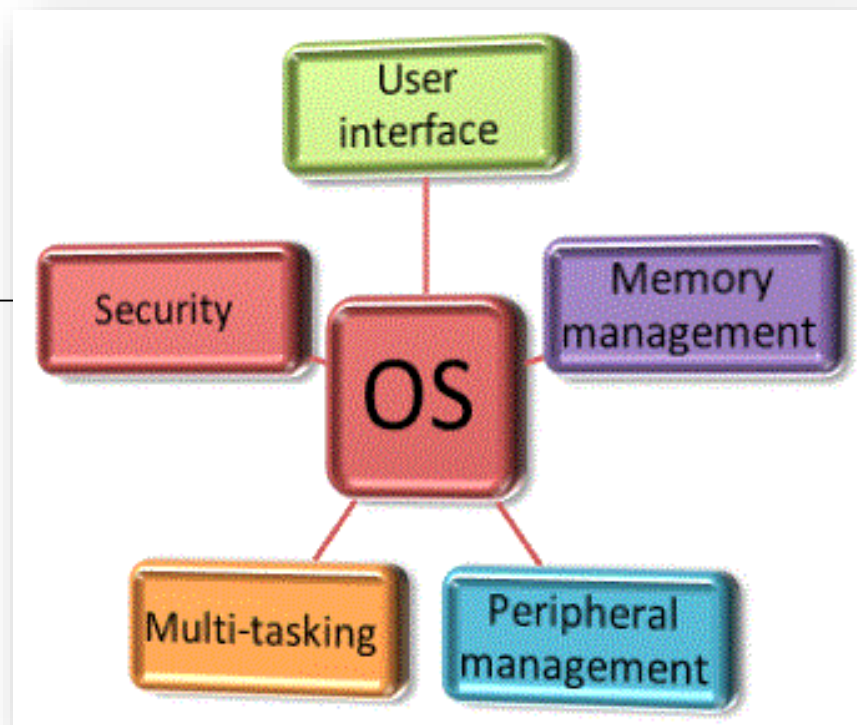
Pliki mogą mieć następujące **atrybuty**:

- tylko do odczytu
- tymczasowy
- ukryty
- archiwalny.

```
1P E7 00 05 D5 CD
2P AB 5E 1E D6 EF
B5 E2 A6 C6 D5 A8
C1 D7 C3 12 00 A9
A1 B7 C5 D4 E3 F6
1P E7 00 05 D5 CD
2P AB 5E 1E D6 EF
B5 E2 A6 C6 D5 A8
C1 D7 C3 12 00 A9
A1 B7 C5 D4 E3 F6
1P E7 00 05 D5 CD
2P AB 5E 1E D6 EF
B5 E2 A6 C6 D5 A8
C1 D7 C3 12 00 A9
A1 B7 C5 D4 E3 F6
1P E7 00 05 D5 CD
2P AB 5E 1E D6 EF
B5 E2 A6 C6 D5 A8
C1 D7 C3 12 00 A9
A1 B7 C5 D4 E3 F6
```

**Fragment pliku binarnego zapisanego  
w kodzie szesnastkowym**

# Zadania realizowane przez system operacyjny



## 3.1. Zadania realizowane przez system operacyjny

### • Komunikacja użytkownika z komputerem

Tę rolę spełnia zewnętrzna warstwa systemu, nazywana **powłoką** (ang. shell), która umożliwia użytkownikowi uruchomienie aplikacji.



Systemy operacyjne z uwagi na komunikację z użytkownikiem możemy podzielić na:

#### a) Systemy tekstowe

```
D:\Documents and Settings\Marcin>dir
Wolumin w stacji D to DANE
Numer seryjny woluminu: 3C08-194D

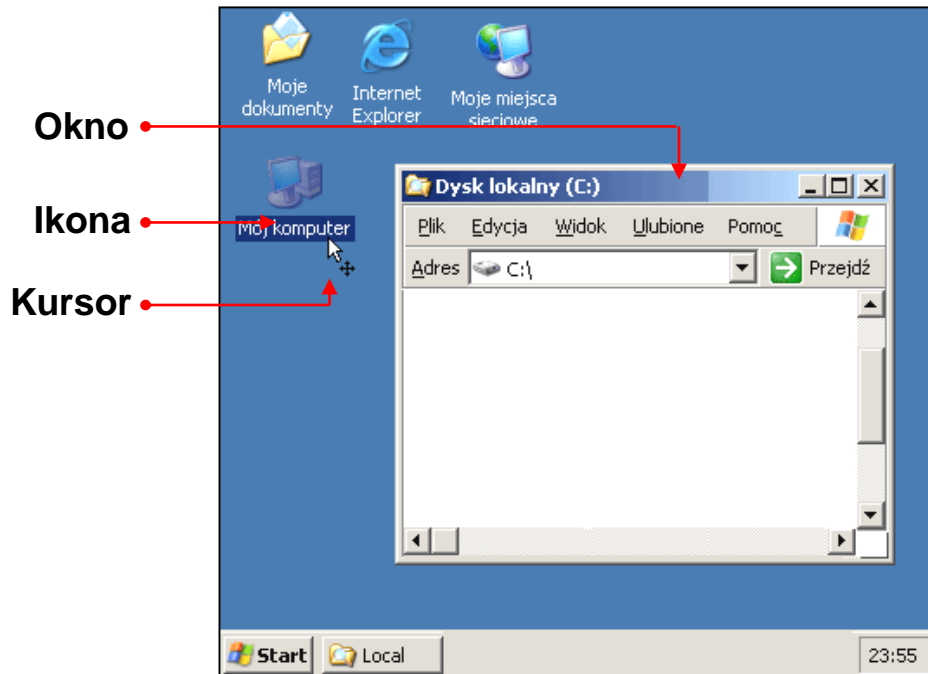
Katalog: D:\Documents and Settings\Marcin

2002-04-05  19:03    <DIR>        .
2002-04-05  19:03    <DIR>        ..
2002-04-05  18:17    <DIR>        Menu Start
2002-04-05  19:04    <DIR>        Recent
2002-04-05  19:04    <DIR>        Moje dokumenty
2002-04-05  19:04    <DIR>        Ulubione
2002-04-05  18:17    <DIR>        Pulpit
2002-04-05  22:54    <DIR>        WINDOWS
2002-05-21  08:11                0 AdobeWeb.log
2002-07-23  22:36                39 968 26_2.tm~
2002-07-23  22:38                26 144 26_3.tm~
2002-07-23  22:38                26 144 26_4.tm~
2002-07-23  22:39                164 384 26_5.tm~
2002-07-23  22:39                164 384 26_6.tm~
2002-08-30  09:29    <DIR>        Gadu-Gadu
2002-10-11  14:25    <DIR>        USWebCache
                6 plik(ów)                421 024 bajtów
                10 katalog(ów)       299 737 088 bajtów wolnych

D:\Documents and Settings\Marcin>
```

Wiersz poleceń

#### b) Systemy graficzne











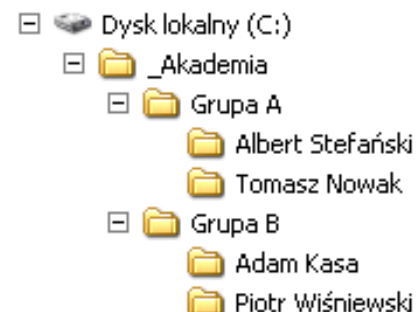
## 3.2. Zadania realizowane przez system operacyjny

### • Zarządzanie plikami

Dane w systemie operacyjnym przechowywane są w postaci plików w urządzeniach zwanych pamięcią masową takich jak: dyski twarde, karty pamięci. System operacyjny umożliwia użytkownikowi zarządzanie plikami dając mu możliwość ich tworzenia, kopiowania, przenoszenia, usuwania i przeglądania.

#### Typowe typy plików wraz z atrybutami

Nazwa	Rozmiar	Typ	Data modyfikacji
 BAZA.MDB	78KB	Baza danych programu MS Access	2002-12-05 00:40
 KOSTKA.BMP	1KB	Obraz - mapa bitowa	2002-12-04 23:39
 KRÓTKITEKST.TXT	1KB	Dokument tekstowy	2002-12-09 16:34
 OFERTA.PPT	381KB	Prezentacja Microsoft PowerPoint	2002-12-01 23:32
 OUTPUT.DBF	1KB	dBASE For Windows	2002-12-03 14:14
 REJESTR.XLS	21KB	Arkusz Microsoft Excel	2002-12-09 10:18
 WIZYTÓWKA.DOC	120KB	Dokument Microsoft Word	2002-12-09 09:50
 ZEGAR.DWG	28KB	AutoCAD Drawing	2002-12-09 10:19



**Pliki mogą być porządkowane, przez umieszczanie ich w drzewiastej strukturze katalogów**

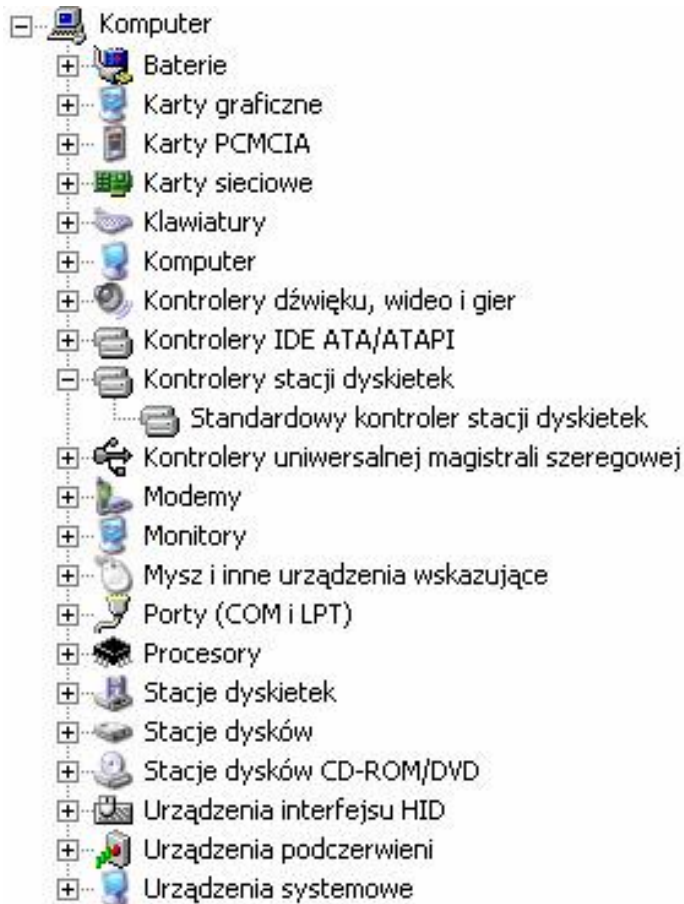
Plik opisany jest za pomocą następujących informacji:

- nazwa pliku oraz rozszerzenie charakterystyczne dla typu pliku,
- rozmiar pliku podany w bajtach (B), kilobajtach (kB), megabajtach (MB), gigabajtach (GB) itd.,
- data oraz godzina utworzenia pliku
- atrybuty, np. „tylko do odczytu”, „ukryty”



## 3.3. Zadania realizowane przez system operacyjny

### • Zarządzanie zasobami maszyny



#### Urządzenia widziane przez system MS Windows XP

Instalowanie urządzenia polega na pobraniu sterownika danego urządzenia, przypisaniu przerwań sprzętowych (ang. IRQ), oraz kanałów do komunikacji urządzeń z pamięcią komputera (ang. DMA).

System operacyjny **obsługuje urządzenia** wchodzące w skład komputera. Wykonywane zadania to:  
Obsługa wewnętrznych i zewnętrznych elementów komputera takich jak: procesor, płyta główna, pamięć lub drukarka, skaner itp.

Badanie stanu urządzeń oraz poprawności komunikacji.

Udostępnianie urządzeń aplikacjom (=użytkownikowi) np. udostępnienie drukarki w edytorze tekstu, skanera w programie graficznym, itp.,  
Odłączanie urządzeń od komputera.



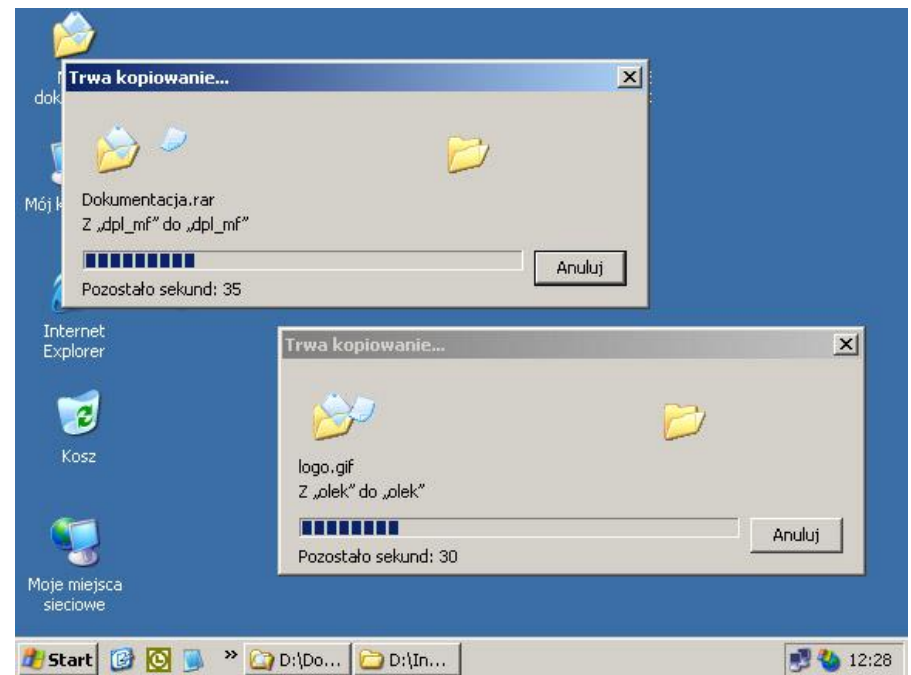
## 3.4. Zadania realizowane przez system operacyjny

### • Uruchamianie aplikacji

System operacyjny nie zawiera programów użytkowych (np. edytorów tekstu, arkuszy kalkulacyjnych, itp.). Każda z aplikacji jest dołączana do systemu operacyjnego poprzez instalację. Aplikacje będące zazwyczaj zbiorem plików, są przez system operacyjny przechowywane w pamięci masowej. System operacyjny umożliwia ich uruchomienie na żądanie użytkownika.

Systemy operacyjne z uwagi na liczbę jednocześnie wykonywanych zadań, np. uruchomionych aplikacji, dzielimy na:

- **Systemy jednozadaniowe.** System może jednocześnie wykonywać tylko jedno zadanie (ciąg instrukcji), (np. MS-DOS).
- **Systemy wielozadaniowe.** System może wykonywać w tym samym czasie kilka zadań, np. nadzorować proces drukowania w czasie edycji tekstu w programie.



Typowym elementem obrazującym wielozadaniowość jest jednocześnie kopiowanie dwóch różnych zbiorów plików

## 3.5. Zadania realizowane przez system operacyjny

---

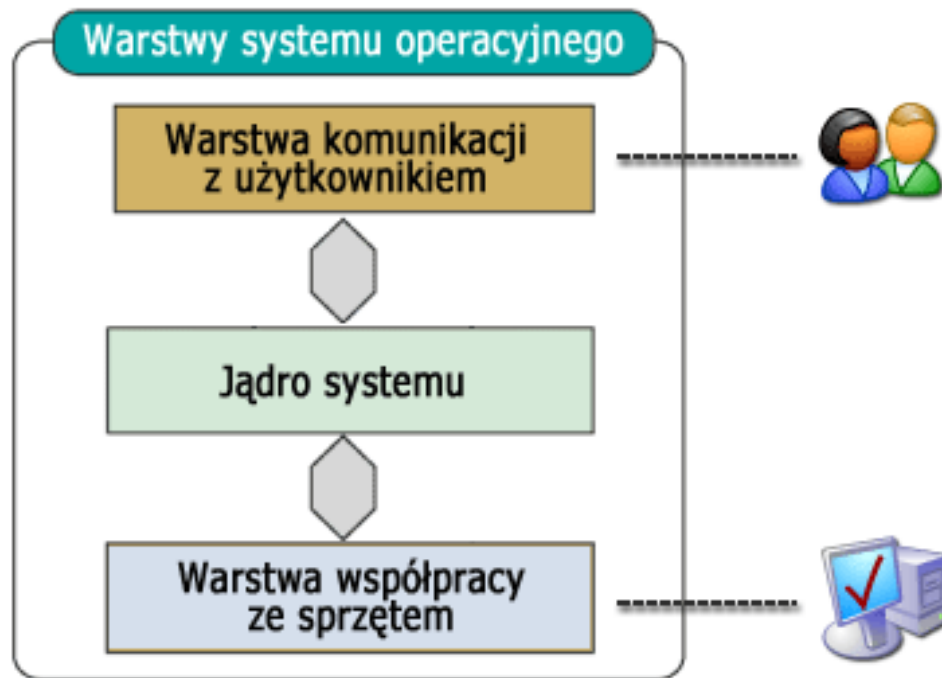
- **Komunikacja z innymi maszynami**

Dzięki modułom systemu operacyjnego odpowiedzialnym za obsługę sieci komputerowych możliwy jest dostęp zarówno do sieci tzw. lokalnych (intranet) jak i globalnych (Internet obejmujący swym zasięgiem cały świat).



## 4. Warstwy systemu operacyjnego

---



W każdym systemie operacyjnym występują mniej lub bardziej wyodrębnione warstwy składające się na architekturę systemu.

W ogólnym modelu systemu operacyjnego można wyszczególnić następujące warstwy przypisując im wyszczególnione zadania:

- powłokę, stanowiącą interfejs użytkownika (komunikacja z użytkownikiem) ,
- jądro systemu realizujące jego funkcje (zarządzanie plikami, uruchamianie aplikacji),
- warstwę odpowiedzialną za współpracę ze sprzętem (zarządzanie zasobami maszyny, komunikacja z innymi maszynami).

# 5.1. Historia systemów operacyjnych - Unix

Lata 60-te – prace na systemem Multisc  
poprzednikiem Unixa

1969 – powstanie systemu Unix

1975 – UNIX edition 5

1975 – 1 BSD

1977 – UNIX edition 6

1978 – 3 BSD

1979 – UNIX edition 7

1982 – UNIX System III

1983 – UNIX System V

Od 1984 – Powstanie odmian systemu Unix takich  
jak: Xenix, AIX, SunOS, Ultrix, HP-UX, Solaris, SVR4

1991 – Linux 0.01

1994 – Linux 1

1994 - do dziś – Kolejne odmiany systemów rodziny

UNIX oraz odmiany Linuxa: Debian, Red HAT,

SlackWare oraz ich następne modyfikacje

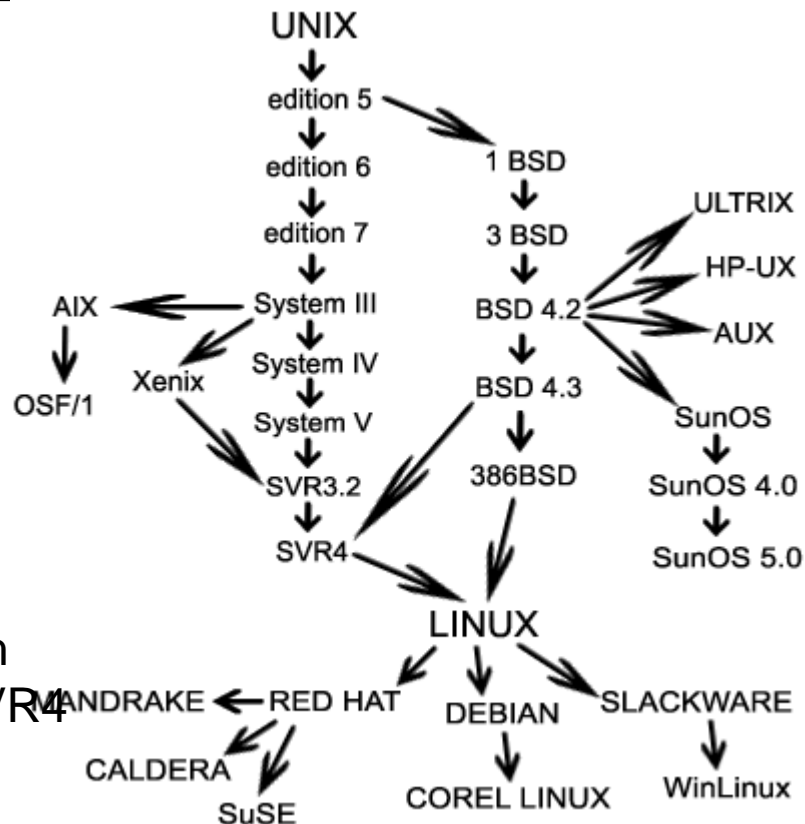


Diagram obrazujący przemiany systemów  
opartych na Unix

## 5.2. Historia systemów operacyjnych – MS Windows

---

1983 – zapowiedź Microsoft Windows  
1985 – prezentacja Microsoft Windows  
1986 – Microsoft Windows 2.03  
1990 – Microsoft Windows 3.0  
1992 – Microsoft Windows 3.1  
1993 – Microsoft Windows 3.11 oraz NT  
1995 – Microsoft Windows 95 oraz NT 3.5  
1996 – Microsoft Windows NT 4.0  
1998 – Microsoft Windows 98  
1999 – Microsoft Windows Milenium  
2000 – Microsoft Windows 2000  
2001 – Microsoft Windows XP  
2003 – Microsoft Windows 2003 server

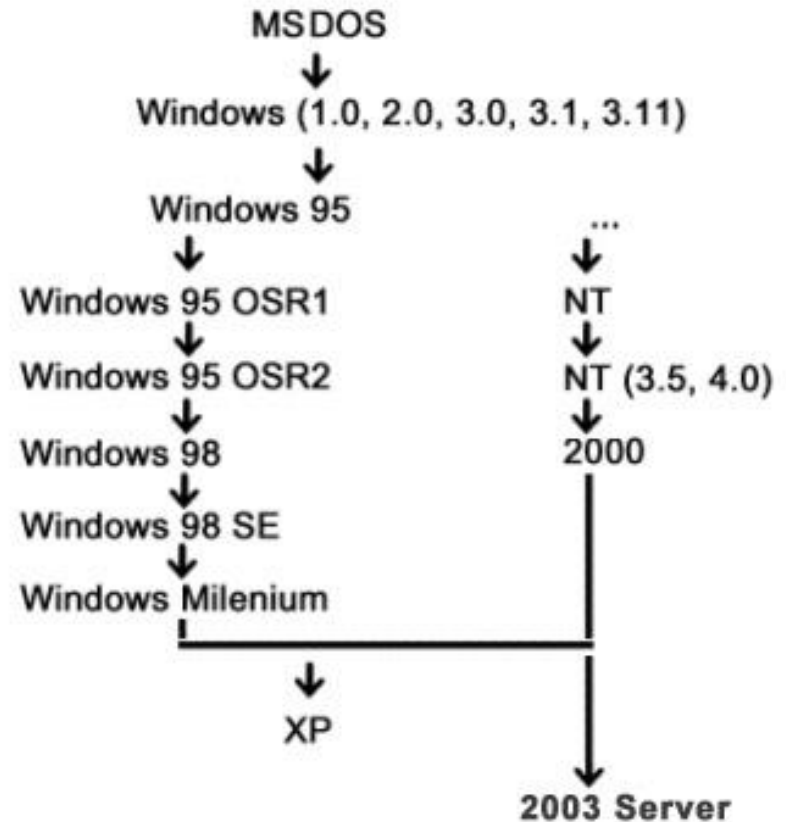


Diagram obrazujący przemiany systemów rodziny Windows

## 6.1. Systemy operacyjne - DOS

**DOS** – (ang. Disk Operating System), czyli Dyskowy System Operacyjny firmy Microsoft. System działa w trybie tekstowym. Wszystkie polecenia wydaje się za pomocą klawiatury.



```
C:\ MS-DOS
(C) Copyright 1985-2001 Microsoft Corp.

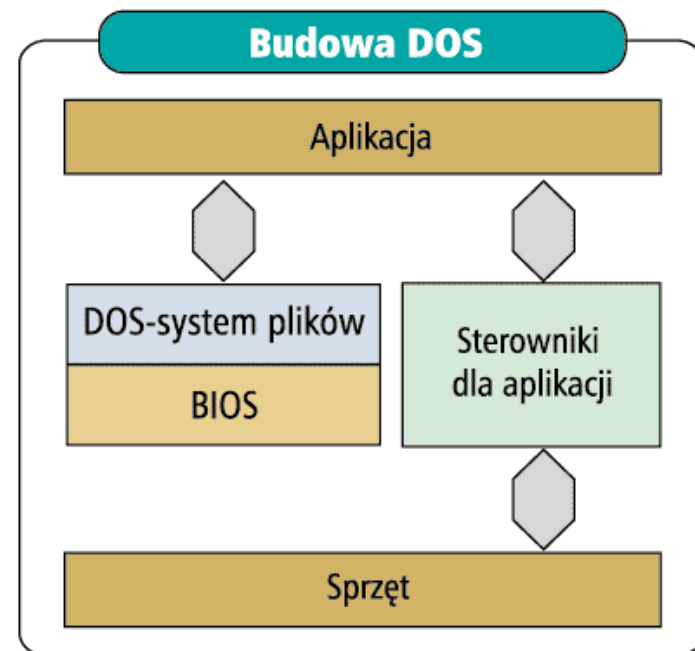
D:\Documents and Settings\Ania>dir
Wolumin w stacji D to SYSTEM_XP
Numer seryjny woluminu: 3E20-336F

Katalog: D:\Documents and Settings\Ania

2003-01-11  15:48    <DIR>        .
2003-01-11  15:48    <DIR>        ..
2003-01-11  15:23    <DIR>        Menu Start
2003-01-11  15:49    <DIR>        Ulubione
2003-01-11  15:23    <DIR>        Pulpit
2003-01-13  12:11    <DIR>        USWebCache
2003-01-05  22:27    <DIR>        WINDOWS
2003-01-30  11:52    <DIR>        Moje dokumenty
2003-02-15  12:54    <DIR>        .javaws
2003-02-15  13:02    <DIR>        .java
2003-02-15  13:02    <DIR>        .jpi_cache
2003-02-18  18:28    <DIR>        Application Data
2003-09-15  20:45    <DIR>        691 .plugin141_01.trace
                1 plik(ów)          691 bajtów
                12 katalog(ów)    1 053 704 192 bajtów wolnych

D:\Documents and Settings\Ania>
```

Widok ekranu z systemem DOS



Budowa systemu DOS

DOS jest systemem **jednozadaniowym**, to znaczy w dowolnej chwili tylko jeden program może pracować pod jego kontrolą.



A:dir				
COMMAND	COM	4896	8-23-83	1:15a
FORMAT	COM	2688	1-01-80	1:01a
RECV	EXE	1024	8-23-83	1:02a
DEBUG	COM	6016	8-22-83	3:05p
CHKDSK	COM	1728	8-22-83	3:00p
FILCOM	COM	8320	8-22-83	3:03p
EDLIN	COM	2432	8-22-83	3:06p
LINK	EXE	41856	8-22-83	3:13p
EXEZBIN	EXE	1280	8-22-83	3:07p
MASM	EXE	70784	8-22-83	3:21p
SYS	COM	608	8-22-83	3:23p
FORMAT	OBJ	4224	8-22-83	3:25p
CREF	EXE	13824	8-22-83	3:02p
LIB	EXE	32128	9-20-83	2:18p
RDCPM	BAK	1920	9-20-83	2:19p
RDCPM	COM	9600	9-20-83	2:20p
RDCPM	OBJ	132	1-01-80	1:04a

17 File(s)

A:

COPY 1

SKP 1

COPY TO

SKP TO

COPY LN

KILL

CANCEL

BACKSPC



C:\PROGRAM FILES\MEDIA MACHINES

Name	Size	Date
	►UP--DIR◀	09.03.15
	►SUB-DIR◀	09.03.15
Studio_2_1	►SUB-DIR◀	09.03.15
mbnails	►SUB-DIR◀	09.03.15

►UP--DIR◀ 09.03.15 21:38

Info

The Norton Commander, Version  
1 July 1998

655 360 Bytes Memory  
560 480 Bytes Free  
2 147 155 968 total bytes on dr  
2 147 155 968 bytes free on dri  
0 files and 4 directories  
use 0 bytes in  
C:\PROGRAM FILES\MEDIA MACHIN

Volume Label : NO NAME  
Serial number: 3E53:10FE

No "dirinfo" file in this direc

ROGRAM FILES\MEDIA MACHINES>

t 2Right 3Name 4Exten 5Time 6Size 7Unsort 8Sync 9Print 1

# Versions of MS DOS

- MS-DOS 1.0 version first and basic operating system and is derived from the system 86-DOS.
- MS-DOS 1.25) DS) Double Sides MS-DOS 1.25 and has allowed to deal with the floppy disk with a two-sided (DS) Double Sides
- MS-DOS. MS-DOS 2.0 and the new improvements were added to regulate the hard drive.
- MS-DOS 3.0 ATTRIB MS-DOS 3.0 and enable the use of high-density disks and added new commands such as his command ATTRIB

- MS-DOS 3.1 MS-DOS 3.1 and allowed to deal with local networks
- MS-DOS 3.2 MS-DOS 3.2 provided to deal with drive-sized 3.5-inch
- MS-DOS 3.3 MS-DOS 3.3 provided to deal with drive-sized 3.5-inch high-density and added a new command
- MS-DOS 4.0 DOS SHELL MEM MS-DOS 4.0 added the possibility of dealing with DOS SHELL ordered MEM and allowed to deal with a larger number of files
-

# version of MS-DOS 5.0 and the fundamental changes

Was in version MS-DOS 5.0 enhancements to the program DOS SHELL and the coordinated use new texts have also been added many improvements to take advantage of [memory](#) and deal with additional memory, and added a new command to recover deleted files or restore [disk](#) to its previous state before the restructuring, have also been added facilities in dealing with the command line is to get help information about any command or command storage and retrieval of past or modified easily

# versions of MS-DOS 6.0 - 6.22

Been in the newer versions of MS Dos add improvements to the previous version of MS-DOS 5.0 and has added a [Microsoft](#) many improvements to take advantage of [Aalmakrp](#) and deal with additional memory, and have added a new command such as SCANDISK and is used to repair the hard drive crashes and ordered DELTREE, which is used to clear the sub-indexes all content.

# MS-DOS

- **Pliki**
- **Nazewnictwo**
- **Polecenia**

# Organizacja zapisu na dysku w DOS

- Plik musi mieć nazwę.
- Nazwa może składać się z 1 do 8 znaków, między którymi nie mogą występować odstępy.
- Nazwa może być poszerzona o człon zwany rozszerzeniem. Rozszerzenie może zawierać od 1 do 3 znaków.
- Rozszerzenie oddzielamy od nazwy kropka (.).
- Ani w nazwie ani w rozszerzeniu nie mogą występować znaki o kodzie dzies.  $\leq 32$  (20H).
- Nie dopuszcza się również znaków:

**. , " \ / [ ] : | < > + = ;**

- Nazwa powinna w miarę możliwości odzwierciedlać zawartość informacji zapisanej w tym zbiorze.

# Rozszerzenia zarezerwowane

Pewne rozszerzenia są zarezerwowane dla zbiorów specjalnego typu.

Pliki wykonywalne:

- **.EXE i .COM** - programy w postaci binarnej,
- **.BAT** - zbiory zawierające polecenia systemu operacyjnego.

Rozszerzen EXE i COM

nie wolno nadawać żadnym zbiorom

- jedynie specjalne programy tworzą zbiory o tych rozszerzeniach

- SYS - plik systemowy
- PAS - programy w Pascalu
- BAS - programy w Basic
- C - -"- C
- BAK - poprzednia wersja
- ASM - programy w assemblerze
- LIB - biblioteki programow
- OBJ - kod wynikowy kompilacji
- TMP - tymczasowe
- TXT - tekstowe
- DAT - dane
- DOC - dokument (edytor WORD)

cd: [ms-dos](#)



# PODSTAWOWE INFORMACJE O DOS

- System operacyjny DOS składa się z kilkudziesięciu modułów. Większość z nich to krótkie programy, część z nich służy do sterowania różnymi elementami komputera.
- Główny moduł systemu operacyjnego DOS to program o nazwie COMMAND.COM, ładowany do pamięci komputera w procesie jego uruchamiania. Program ten realizuje podstawowe polecenia systemu operacyjnego DOS, zwane poleceniami wewnętrznymi.
- Jeżeli zlecimy komputerowi - programowi DOS- wykonanie jakiegoś zadania, którego nie może zrealizować COMMAND.COM, to wczytuje on (COMMAND.COM) z dysku odpowiedni moduł systemu operacyjnego (odpowiedni program - polecenie zewnętrzne), który wykona to zadanie. Po wykonaniu moduł ten (program) jest usuwany z pamięci.

# Budowa systemu operacyjnego DOS

## System operacyjny DOS tworzą programy:

- moduł **BIOS**, umieszczony w pamięci ROM, który zapewnia obsługę podstawowych operacji we/wy i wczytuje pozostałą część systemu z dysku
- pliki **IO.SYS**, **MSDOS.SYS**, **COMMAND.COM**,
- polecenia zewnętrzne systemu - pliki dyskowe (na ogół w katalogu DOS)
- 2 dodatkowe zbiory, tworzone przez użytkownika: **CONFIG.SYS**  
i **AUTOEXEC.BAT** (muszą być w katalogu głównym)

## IO.SYS

plik z procedurami obsługi urządzeń, ładowany podczas uruchamiania systemu. Moduł podstawowych operacji we/wy. Pozwala na dołączenie właściwego systemu do modułu BIOS. Zawiera też programy obsługi standardowych urządzeń (klawiatura, monitor, dysk, drukarka)

## MSDOS.SYS

zapewnia współpracę z programami użytkowymi, m.in. zarządza plikami, pamięcią i procesami towarzyszącymi realizacji programów. Plik z procedurami obsługi odwołań systemowych, ładowany podczas uruchamiania systemu. Jadro systemu operacyjnego. Umożliwia komunikację między programami użytkowymi a systemem operac. Zarządza plikami dyskowymi, buforami dyskowymi, wczytuje Command.com. Właściwy DOS - interpretacja poleceń z innych programów i przekazywanie do BIOS.

## COMMAND.COM

plik z interpretatorem poleceń wewnętrznych, ładowany podczas uruchamiania systemu oraz po wykonaniu każdego polecenia, które wymazało go z pamięci. Składa się z 2 części: rezydentnej i ładowalnej.

## config.sys

plik tekstowy, opisujący konfigurację systemu, interpretowany podczas uruchamiania systemu

## autoexec.bat

plik z poleceniami, wykonywanymi w trybie wsadowym, bezpośrednio po uruchomieniu systemu

# Symbole grupowe

- \* - zastępuje dowolną grupę znaków
- ? - zastępuje pojedynczy znak

# Nazwy urządzeń

- A:, B: - stacje dyskietek
- C:, D:... - dyski (partycje dysków, wolumeny sieciowe)
- NUL - urządzenie puste
- CON - konsola (klawiatura przy czytaniu, ekran przy pisaniu)
- PRN, LPT1 - drukarka pierwsza
- LPT2 - drukarka2
- AUX, COM1 - złącze szeregowe 1
- COM2 - złącze szeregowe 2

# Strumienie

**>plik, >urządzenie** - wypisanie wyników do podanego pliku/urządzenia,  
skasowanie starej zawartości

**>>** - dopisanie wyników do pliku

**... | ....** - potok

# Polecenia DOS

CLS - Wymazanie ekranu i umieszczenie kursora w lewym górnym rogu ekranu

A: - Dysk A: staje się dyskiem bieżącym

C: - Dysk C: -"-

VER - Wyświetlenie wersji systemu operacyjnego, np.

DATE - Wyświetlenie daty bieżącej systemowej.

TIME - Wyświetla aktualny czas, z możliwością zmiany.

VOL - Wyświetlenie na ekranie nazwy bieżącego dysku

FORMAT A:

DIR - (polecenie wewnętrzne)

# Katalogi dyskowe

Spis treści dysku nazywamy **k a t a l o g i e m**. Może on tworzyć strukturę drzewa. Porównując spis treści książki z katalogiem dysku widzi się wiele podobieństw.

Przykładowe drzewo katalogów:

A:.

|-----DOS

|-----EDYTORY

|     |----WORD

|             PODANIE.DOC

|     |----TAG

|             LISTA1.TAG

|             LISTA2.TAG

|-----ARKUSZ

|     |----EXCEL

|     |----LOTUS

|-----JEZYKI

|     |----TP

|     |----QBASIC

# Ścieżka dostępu (path)

- droga jaka należy przebyć z katalogu głównego, by dotrzeć do pliku.

Np. ścieżka dostępu do LIST1.TAG:

A:\EDYTORY\TAG\LIST1.TAG

do pliku Podanie.tag:

A:\EDYTORY\WORD\PODANIE.TAG

Nazwy katalogów oddzielamy od siebie znakiem \ (backslash).

\ użyty na początku, oznacza katalog główny, zwany korzeniem (root).

Jeżeli ścieżka dostępu nie jest wyświetlana, to należy napisać polecenie:

PROMPT \$p\$g



# Tworzenie katalogów

MD katalog lub MKDIR katalog (polecenie wewnętrzne)

Skrot od Make Directory, czyli stwórz katalog.

np. MD DOS

MD EDYTORY

MD JEZYKI

MD \EDYTORY\TAG - tworzenie podkatalogu TAG w podkatalogu Edytory

MD A:\EDYTORY\TAG - "-" na dysku A:

Inna metoda jest wejście do katalogu a potem stworzenie podkatalogu.

# Poruszanie się po katalogach

CD katalog lub CHDIR katalog (polecenie wewnętrzne) Skrot od Change Directory, czyli zmien katalog, np.

CD DOS CD EDYTORY

CD

CD .. - wejscie do katalogu poprzedzajacego (parent)

CD \ lub CD\ - powrot do katalogu glownego

Format: CD [[d:]path]  
lub  
CHDIR [[d:]path]

Opis: Polecenie umożliwia zmianę katalogu bieżącego na urządzeniu wyspecyfikowanym lub wyświetlenie aktualnej drogi-do- zbioru, o ile pominięte zostały parametry.

Katalog bieżący jest katalogiem, w którym system poszukuje zbiorów, których specyfikacje nie określają w sposób jawny drogi-do-zbioru.

Przykłady: CD \

Polecenie przywraca jako katalog bieżący katalog systemowy.

CD B:\POZIOM1\POIOM2

Powyższy przykład ilustruje zmianę katalogu bieżącego urządzenia b: na

systemowy -> POZIOM1 -> POZIOM2.

CD POZIOM3

Katalog bieżący dla urządzenia bieżącego jest w przykładzie zmieniany na katalog niższego poziomu POZIOM3 katalogu bieżącego.

# Usuwanie katalogu

RD katalog lub RMDIR katalog (polec. wewnętrzne) Skrot od Remove Directory - usun katalog

Katalog usuwany nie moze zawierac zbiorow.

Format: RD [d:]path czyli RD [d:]sciezka  
lub  
RMDIR [d:]path

Opis: Usuwa wyspecyfikowany katalog podrzedny. Ostatni wyspecyfikowany w drodze-do-zbioru katalog jest usuwany. Katalog taki

nie moze zawierac zadnych zbiorow. Katalogi: systemowy i biezacy nie moga byc usuniete.

Polecenie nie dziala z SUBST.

Przyklady: RMDIR C:POZIOM1

Usuwa katalog podrzedny POZIOM1. Usuwany katalog jest pusty

i nie jest katalogiem biezacym.

RM \POZIOM1\POZIOM2

Usuwa katalog podrzedny POZIOM2 z katalogu POZIOM1 na urzadzeniu biezacym. Katalog POZIOM2 nie moze zawierac zadnych zbiorow.

# Wyswietlanie i kopiowanie plikow

TYPE skad\co

*(polecenie wewnetrzne)*

TYPE plik | MORE

MORE < plik

# **Pliki wsadowe**

# Plik wsadowy (ang. *batch file*)

program komputerowy wykonywany w trybie wsadowym (tj. bez wpływu użytkownika na przebieg programu) zapisany w pliku.

Programy wsadowe mogą być wykonywane w różnych systemach operacyjnych, w szczególności w:  
**Unix, Linux, DOS, Windows.**

Programem wsadowym może być kompilowany program komputerowy lub skrypt.

# Plik wsadowy (ang. *batch file*)

Program wsadowy z założenia nie oczekuje współpracy (interakcji) z użytkownikiem w trakcie wykonywania. Po uruchomieniu programu użytkownik nie ma wpływu na przebieg programu wsadowego, może jedynie oczekiwać na jego zakończenie lub ewentualnie przerwać jego działanie. Jednak jeśli program wsadowy wymaga podania parametrów potrzebnych do działania, np. opcji, nazwy pliku, katalogu czy nazwy użytkownika, należy podać je w chwili uruchamiania programu w wierszu poleceń jako argumenty wywołania programu. Często z programami wsadowymi utożsamiane są skrypty, ale nie wszystkie skrypty wykonywane są w trybie wsadowym, wiele skryptów działa bowiem w trybie interakcyjnym, tj. oczekując współdziałania użytkownika.

# Plik wsadowy (ang. *batch file*)

Program wsadowy z założenia nie oczekuje współpracy (interakcji) z użytkownikiem w trakcie wykonywania. Po uruchomieniu programu użytkownik nie ma wpływu na przebieg programu wsadowego, może jedynie oczekiwać na jego zakończenie lub ewentualnie przerwać jego działanie. Jednak jeśli program wsadowy wymaga podania parametrów potrzebnych do działania, np. opcji, nazwy pliku, katalogu czy nazwy użytkownika, należy podać je w chwili uruchamiania programu w wierszu poleceń jako argumenty wywołania programu. Często z programami wsadowymi utożsamiane są skrypty, ale nie wszystkie skrypty wykonywane są w trybie wsadowym, wiele skryptów działa bowiem w trybie interakcyjnym, tj. oczekując współdziałania użytkownika.



# Programy wsadowe w systemach MS-DOS i MS Windows

- W systemach MS-DOS i Windows typowym programem wsadowym jest plik tekstowy zawierający ciąg poleceń interpretera komend, z angielskiego nazywany batchem lub plikiem batchowym (ang. *batch file*); w środowisku uniksowym taki plik (zawierający ciąg poleceń powłoki systemowej) nazywany jest skryptem. Po uruchomieniu programu wsadowego interpreter czyta z pliku i wykonuje kolejne zapisane w nim polecenia.
- Programy wsadowe systemu MS-DOS posiadają rozszerzenia .BAT lub .CMD. AUTOEXEC.BAT to specjalny program wsadowy uruchamiany podczas startu systemu.
- Interpreterem komend (powłoką) w systemach MS-DOS i Windows 95/98 jest program command.com, w systemach Windows z linii NT (Microsoft Windows NT, 2000, XP, 2003, Vista) – program cmd.exe.

# Przykładowe komendy wsadowe

będące poleceniami powłoki (interpretera komend) systemu DOS i Windows:

**ECHO** - wyświetlanie tekstu na ekranie

**COPY** - kopiowanie pliku

**DEL** - usuwanie pliku

**PROMPT** - ustalenie wyglądu zachęty

**REM** - komentarz (instrukcja niewykonywana)

**GOTO** etykieta (skok do linii oznaczonej etykietą)

# Przykładowy program wsadowy

systemu DOS lub Windows, w tym przykładzie wyświetlający napis *Hello World* :

```
@echo off
```

```
echo Hello World!
```

```
pause
```

```
:end
```

TT.BAT

Wykonanie skryptu polega na napisaniu polecenia:  
> TT

doskey n=notepad \$1  
doskey n=notepad \$1.bat

# Przykładowy skrypt

proszący o podanie imienia i wypisujący stosowny komunikat:

```
@echo off
```

```
set /p imie=Jak masz na imię?
```

```
echo Witaj %imie%!
```

```
pause
```

```
:end
```

Polecenie SET z parametrem "/P" rejestruje wszystkie znaki które wprowadzimy w konsoli i przypisze je zmiennej (tu: var):

```
SET /P var=
```

# Tworzenie kopii pliku **hi.bat** do **hi.bak**

**@ECHO OFF**

**CB.BAT**

**IF NOT EXIST** hi.txt **GOTO** BrakPliku

**IF** **EXIST** hi.bak **GOTO** JestKopia

**ECHO** Kopiowanie hi.txt na hi.bak...

**COPY** hi.txt hi.bak > **NUL**

**ECHO** Zadanie wykonane...

**GOTO** Koniec

:BrakPliku

**ECHO** Plik hi.txt nie istnieje...

**GOTO** Koniec

: JestKopia

**ECHO** Plik hi.bak już istnieje...

:Koniec

# Argumenty plików wsadowych

Pliki wsadowe można uruchamiać z argumentami. Do kolejnych argumentów pliku odwołujemy się za pomocą symboli *hi*, %2, ..., %9, które reprezentują pierwszy, drugi, ..., dziewiąty argument pliku. Przykład:

**@ECHO OFF**

**IF NOT EXIST %1.txt GOTO BrakPliku**

**IF EXIST %1.bak GOTO IstniejeKopia**

**ECHO** Kopiowanie %1.txt na %1.bak...

**COPY %1.txt %1.bak > NUL**

**ECHO** Wykonane...

**GOTO Koniec**

:BrakPliku

**ECHO** Plik %1.txt nie istnieje...

**GOTO Koniec**

:IstniejeKopia

**ECHO** Plik %1.bak już istnieje...

:Koniec

**CB2.BAT**

# Pliki wsadowe – polecenia ECHO i SET

Aby **utworzyć** zmienną należy użyć komendy SET:

> SET liczba=17

> SET wyraz=Ada

Aby **wyświetlić** zawartość zmiennej

> echo %liczba%

> echo %wyraz%

```
A:\>set liczba=17
```

```
A:\>set wyraz=Ada
```

```
A:\>echo %wyraz% %liczba%  
Ada 17
```

> Aby zwiększyć zmienną o 1 nie można po prostu wpisać:

> SET zmienna=1

> SET zmienna=%zmienna%+1

> ECHO %zmiennna%

Aby nie traktowało tego jako napis a jako wartość policzalną należy do polecenia SET uruchomić przełącznikiem /A

> SET /A zmienna=%zmienna%+1

# Pliki wsadowe – polecenia ECHO i SET

W zmiennych jest możliwość podmieniania ich części:

%zmienna:nap1=nap2%

```
A:\>set str=ABCDcba
A:\>echo %str:A=X%
XBCDCbX
```

Parametr po "~" oznacza od której litery w ciągu ma zacząć, a po przecinku ile liter ma zostać wyświetlonych. Brak drugiego parametru - oznacza „do końca tekstu”

```
A:\>set im=Adam Czartoryski
A:\>echo %im:~5%
Czartoryski
A:\>echo %im:~5,7%
Czartor
```

Polecenie SET uruchomione z parametrem /P powoduje rejestrowanie WE i zapamiętanie wprowadzonych znaków i przypisaniu do zmiennej:

**SET /P zmienna=**



# Batch files



# Batch files

allow MS-DOS & Windows users to create a **lists of commands to run** in sequence once the batch file has been executed.

For example, a batch file could be used to run frequently run commands, deleting a series of files, moving files, etc. A simple batch file does not require any special programming skills and can be done by users **who have a basic understanding of MS-DOS commands.**

# Batch files – examples (1)

```
@echo off
```

```
cls
```

```
echo Hello this a test batch file
```

```
pause
```

```
dir c:\windows\*.log
```

# Batch files – examples (2)

## Sample Script with Comments

```
rem Creation Date: 2/28/2015                Last Modified: 3/15/2015
rem Author: William R. Stanek
rem =====
rem  Description:
rem      Displays system configuration information including system name,
rem      IP configuration and Windows version.
rem =====
rem  Files: Stores output in sys-info.txt.
rem =====
```

```
hostname    > sys-info.txt
ver         >> sys-info.txt
systeminfo >> sys-info.txt
ipconfig -all >> sys-info.txt
type        sys-info.txt
```

# Batch files – examples (2')

```
rem Creation Date: 2/28/2015
rem Author: William R. Stanek
rem =====
rem Files: Stores output in sys-info.txt.
rem =====
```

```
hostname    > sys-info.txt
ver         >> sys-info.txt
systeminfo >> sys-info.txt
ipconfig -all >> sys-info.txt
type sys-info.txt
```

Last Modified: 1/10/2018

*Zmodyfikuj ten plik, by  
pokazywał TYLKO:*

-----  
Nazwę komputera  
Wersję systemu  
BIOS  
RAM (wielkość)  
Adres IP  
Maska  
Brama

# Batch commands

@	Does not echo back the text after the at symbol. This most often used as <b>@ECHO OFF</b> to prevent any of the commands in the batch file from being displayed, just the information needed.
%1	<p>The percent followed by a numeric value, beginning with one, allows users to add variables within a batch file. The below line is an example of what can be used in a batch file.</p> <p><b>ECHO Hello %1</b></p> <p>When the above one-line batch file is created, add your name after the batch file. For example, typing myname (being the name of the bat file) and then your name:</p> <p><b>myname Bob</b></p> <p>would output:</p> <p>Hello Bob</p> <p><b>Note:</b> This can be extended to %2, %3, and so on.</p>

# Batch commands

<b>::</b>	One of two ways of adding <a href="#">remarks</a> into the batch file without displaying or executing that line when the batch file is run. Unlike REM, this line will not show regardless if ECHO off is in the batch file.
<b>:LABEL</b>	By adding a colon in front of a word, such as <b>LABEL</b> , you create a category, more commonly known as a <a href="#">label</a> . This allows you to skip to certain sections of a batch file such as the end of the batch file. Also see <b>GOTO</b> .
<b>CALL</b>	This used to run another batch file within a batch file.

# Batch commands

<b>ECHO</b>	Will echo a message in the batch file. Such as <b>ECHO Hello World</b> prints <i>Hello World</i> on the screen when executed. However, without <b>@ECHO OFF</b> at the beginning of the batch file you'll also get "ECHO Hello World" and "Hello World." Finally, if you'd just like to create a blank line, type <b>ECHO.</b> adding the period at the end creates an empty line.
<b>EXIT</b>	Exits out of the DOS window if the batch file is running from Windows.
<b>GOTO LABEL</b>	Used to go to a certain label, such as LABEL. An example of GOTO would be to GOTO END..
<b>START</b>	Used for <a href="#">Win 95+</a> to start a Windows application; such as <b>START C:\WINDOW\CALC</b> would run the Calculator.



# Batch files - examples

```
@echo off
if "%1" == "" goto error
echo Hello %1, it's nice to meet you!
goto end
:error
echo Type your name after batch file!
:end
```

**echo Y | del \*.\*     Answers the DEL "Are you sure" question automatically**

**echo %PATH%             Displays the value of PATH, the current search path**

# Batch files - examples

Instrukcja iteracyjna *FOR* pliku wsadowego jest pętlą o znanej liczbie powtórzeń i następującej składni:

## **FOR %%X IN (test) DO komenda**

Argument *test* jest obowiązkowy i zawiera pooddzielane znakami odstępu nazwy plików - w nazwach tych mogą występować znaki wieloznaczne '?' i '\*'. **Po znakach '%%' podajemy zmienną.** Zmienna ta będzie kolejno przybierać wszystkie możliwe wartości spośród wartości podanych na liście (*test*). Dla każdej możliwej wartości zmiennej X zostanie wykonane polecenie *komenda*.

Przykłady:

```
FOR %%A IN (1 2 3) DO DIR *.txt
```

```
FOR %%K IN (zadanie zadanie.bat zadanie.bak) DO DIR
```

```
FOR %%f IN (*.txt) DO type %%f
```

# Batch files - examples

Korzystając m.in. z poleceń **find** oraz **sort** napisać plik wsadowy **mydir.bat**, który wyświetli zawartość bieżącego katalogu.

Na wydruku powinny być widoczne tylko pliki i katalogi.

Wydruk powinien zawierać w pierwszej kolejności listę plików a następnie listę katalogów.

Zarówno pliki jak i katalogi należy posortować alfabetycznie.

# How you can implement the choice options

- `@ECHO OFF`  
`REM - LABEL INDICATING THE BEGINNING OF THE DOCUMENT.`  
`:BEGIN`  
`CLS`  
`REM - THE BELOW LINE GIVES THE USER 3 CHOICES (DEFINED AFTER /C:)`  
`CHOICE /N /C:123 /M "PICK A NUMBER (1, 2, or 3)"test`  
`REM - THE NEXT THREE LINES ARE DIRECTING USER DEPENDING UPON INPUT`  
`IF ERRORLEVEL ==3 GOTO THREE`  
`IF ERRORLEVEL ==2 GOTO TWO`  
`IF ERRORLEVEL ==1 GOTO ONE`  
`GOTO END`  
`:THREE`  
`ECHO YOU HAVE PRESSED THREE`  
`GOTO END`  
`:TWO`  
`ECHO YOU HAVE PRESSED TWO`  
`GOTO END`  
`:ONE`  
`ECHO YOU HAVE PRESSED ONE`  
`:END`

# How to start Windows files and other programs from a batch file

> START /MAX C:\Windows\NOTEPAD.EXE

# How to make a time log in a batch file

```
ECHO. |TIME > TIME  
COPY LOG +TIME
```

Another alternative is:

```
echo. |time |find "current" >> log
```

# FIND command

- search for a specific string in a file(s) & send the specified lines to output device.

Syntax:

**FIND [/V or /C][/I][/N] "string" [drive:][path]filename**

where:

/V	Displays all lines not containing the string specified.
/C	Displays the count of lines containing the string.
/I	Ignores the case of characters when searching for the string.
/N	Displays the line numbers with the displayed lines.
/OFF[LINE]	Do not skip files with offline attribute set ( <b>only available in Windows XP and later versions</b> ).
"string"	Specifies the text string to find.
drive:\path	Specifies the location of the file or files to search.
filename	Specifies the name of the file to be searched.

# FIND command

With /C, FIND may be used for counting.

Use the FIND command to check if your HTML files have a closing tag for each opening tag:

```
C:\>FIND /C /I "<TD" example.html
```

```
----- example.html: 20
```

```
C:\>FIND /C /I "</TD" example.html
```

```
----- example.html: 20
```



# FINDSTR searches for strings in files.

FINDSTR [/B] [/E] [/L] [/R] [/S] [/I] [/X] [/V] [/N] [/M] [/O] [/P] [/F:*file*] [/C:*string*] [/G:*file*]  
[ /D:*dir list*] [/A:*color attributes*] [*strings*] [[*drive:*][*path*]*filename*[ ...]]

/B	Matches pattern if at the beginning of a line.
/E	Matches pattern if at the end of a line.
/L	Uses search strings literally.
/R	Uses search strings as regular expressions.
/S	Searches for matching files in the current directory and all subdirectories.
/I	Specifies that the search is not to be case-sensitive.
/X	Prints lines that match exactly.
/V	Prints only lines that do not contain a match.
/N	Prints the line number before each line that matches.
/M	Prints only the filename if a file contains a match.
/O	Prints character offset before each matching line.
/P	Skip files with non-printable characters.
/A: <i>attr</i>	Specifies color attribute with two hex digits. See "color /?"
/F: <i>file</i>	Reads file list from the specified file(/ stands for console).
/C: <i>string</i>	Uses specified string as a literal search string.
/G: <i>file</i>	Gets search strings from the specified file(/ stands for console).
/D: <i>dir</i>	Search a semicolon delimited list of directories.
<i>strings</i>	Text to be searched for.

# FINDSTR Regular expression

.	Wildcard: any character
*	Repeat: zero or more occurrences of previous character or class
^	Line position: beginning of line
\$	Line position: end of line
[ <b>class</b> ]	Character class: any one character in set
[ <b>^class</b> ]	Inverse class: any one character not in set
[ <b>x-y</b> ]	Range: any characters within the specified range
\x	Escape: literal use of metacharacter x
\< <b>xyz</b>	Word position: beginning of word
<b>xyz</b> \>	Word position: end of word

# Using the **For** Command

The **For command** fulfills a special niche in batch file programming. You know that you can use wildcard characters to make multiple file selections when needed. Unfortunately, using wildcard characters won't always work. Sometimes you need to know the name of the file. A command line utility might not support wildcard characters or the file argument doesn't easily fit within the wildcard method of description. That's where the For statement comes into play for batch files. The form is:

**FOR %%variable IN (set) DO command [parameters]**

You can also use this command at the command prompt to process files manually. Instead of using a single percent (%) symbol, you use two in front of the variable. Here's a sample:

**Echo Off**

**For %%F In (\*.BAT \*.TXT) Do Dir %%F /B**

**Echo On**

In this case, the For command processes all of the files that have a BAT or TXT extension in the current directory. The command processes the files in the order in which they appear in the directory and you have no guarantee what the order is. **The %%F variable contains the name of an individual file.** The Dir command is called once for each file with the %%F variable as an input.

# Deleting files equal to 0 in a batch file

```
for /F %%A in ("pics.txt") do if %%~zA equ 0 del pics.txt
```

Another alternative is:

```
FOR %%F IN (*.*) DO (IF %%~zF LSS 1 DEL %%F)
```

**%~zF**

Obtains the size of the input file.

# Zadania

Wszystkie pliki wsadowe powinny działać we wszystkich możliwych przypadkach związanych z istnieniem plików. Należy również zadbać o to, aby podczas wykonywania plików wyświetlane były **wyłącznie** informacje wypisywane przez polecenie ECHO pliku wsadowego.

1. Korzystając m.in. z poleceń *find* oraz *sort* **napisać plik wsadowy *mydir.bat*, który wyświetli zawartość katalogu bieżącego**. Na wydruku powinny być widoczne tylko pliki i katalogi. Wydruk powinien zawierać w pierwszej kolejności listę plików a następnie listę katalogów. Zarówno pliki jak i katalogi należy posortować alfabetycznie. **Nie wolno** korzystać z żadnych opcji polecenia *dir*, można natomiast skorzystać z tymczasowych plików pomocniczych.
2. Napisać plik wsadowy *mycopy.bat* służący do tworzenia kopii pliku pod inną nazwą, przyjmując następujące założenia:
  1. plik działa tylko w przypadku gdy został uruchomiony z dwoma różnymi argumentami,
  2. jeżeli plik o takiej nazwie jak nazwa kopii już istnieje, to nie wykonujemy kopiowania.
3. Napisać plik *myren.bat* służący do zmiany nazwy na nazwę podaną, przyjmując następujące założenia:
  1. plik działa tylko w przypadku gdy został uruchomiony z dwoma argumentami,
  2. jeżeli oba argumenty są takie same, to jeśli plik istnieje nie wykonujemy żadnych czynności, jeśli natomiast pliku nie ma wypisujemy komunikat o błędzie,
  3. jeżeli plik o takiej nazwie jak nowa nazwa pliku już istnieje, to wypisujemy komunikat o błędzie.

# Zadania

1. Napisać plik wsadowy *mycopy.bat* służący do tworzenia kopii pliku pod inną nazwą, przyjmując następujące założenia:  
plik może być uruchomiony z dowolną ilością argumentów,
  1. jeżeli nie podano argumentów wypisujemy składnię,
  2. jeżeli pierwszy argument jest równy "/" wypisujemy informację o przeznaczeniu i składnię,
  3. jeżeli podano argumenty i pierwszy z nich nie jest równy "/" to próbujemy wykonać kopiowanie gdy podano dokładnie dwa różne argumenty, w przeciwnym wypadku (gdy podano więcej niż dwa, gdy podano jeden argument lub gdy argumenty są dwa, ale takie same) wypisujemy komunikat o błędzie,
  4. jeżeli plik o takiej nazwie jak nazwa kopii już istnieje, to nie wykonujemy kopiowania.
2. Napisać plik wsadowy *myren.bat* służący do zmiany nazwy pliku pod podaną, przyjmując następujące założenia:
  1. plik może być uruchomiony z dowolną ilością argumentów,
  2. jeżeli nie podano argumentów wypisujemy składnię,
  3. jeżeli pierwszy argument jest równy "/" wypisujemy informację o przeznaczeniu i składnię,
  4. jeżeli podano argumenty i pierwszy z nich nie jest równy "/" to próbujemy zmienić nazwę gdy podano dokładnie dwa argumenty, w przeciwnym wypadku (gdy podano więcej niż dwa, gdy podano jeden argument lub gdy argumenty są dwa, ale takie same) wypisujemy komunikat o błędzie,
  5. jeżeli plik o takiej nazwie jak nazwa kopii już istnieje, to usuwamy go, a następnie wykonujemy zmianę nazwy pliku,
  6. jeżeli argumenty są dwa i są takie same oraz plik istnieje, to efekt powinien być taki, jakgdyby zmieniono nazwę pliku.