

Skrypt nr 1

Zmiana nazw wszystkich plików w zadanym katalogu (parametr wywołania skryptu), do których mamy ustawione prawo zapisu, przez dopisanie dodatkowego członu **.old**. Wcześniej należy skasować wszystkie pliki, które już mają takie rozszerzenie.

```
#!/bin/sh

if [ "$#" != "1" ]; then
    echo "Skrypt uruchamiany z parametrem (sciezka do katalogu)"
    exit 1
fi

if [ ! -d $1 ]; then
    echo "Nie ma takiego katalogu"
    exit 2
fi

rm -rf $1/*.old
for file in `ls $1/`; do
    if [ -w $1/$file ]; then
        mv $1/$file $1/$file.old
    fi
done
```

Skrypt nr 2

Tworzenie nowych plików w zadanym katalogu (parametr wywołania skryptu), według listy umieszczonej w pliku (drugi parametr wywołania skryptu). Nowe pliki mają być zerowej wielkości (puste). Jeżeli jakiś plik już istnieje, to nie powinien zostać zniszczony.

```
#!/bin/sh

if [ "$#" != "2" ]; then
    echo "Skrypt uruchamiany z dwoma parametrami"
    exit 1
fi

if [ ! -d $1 ]; then
    echo "Nie ma takiego katalogu"
    exit 2
fi

if [ ! -r $2 ]; then
    echo "Nie mozna czytac z pliku lub plik nie istenieje"
    exit 3
fi

for file in `cat $2`; do
    if [ -w "$1" ]; then
        if [ ! -e "$1/$file" ]; then
            touch "$1/$file"
        fi
    fi
done
```

Skrypt nr 3

Kasowanie wszystkich plików w zadanym katalogu (parametr wywołania skryptu), poza plikami wykonywalnymi, mającymi ustawiony bit dostępu typu „execute”.

```
#!/bin/sh

if [ "$#" != "1" ]; then
    echo "Skrypt uruchamiany z parametrem (sciezka do katalogu)"
    exit 1
fi

if [ ! -d $1 ]; then
    echo "Nie ma takiego katalogu"
    exit 2
fi

for file in `ls "$1/"`; do
    if [ ! -x "$1/$file" ]; then
        rm -rf "$1/$file"
    fi
done
```

Skrypt nr 4

Numerowanie wszystkich plików w zadanym katalogu (parametr wywołania skryptu), do których mamy ustawione prawo wykonywania („execute”), przez dodanie dodatkowego członu rozszerzenia o postaci .jnumer kolejny. Numeracja powinna przebiegać według wielkości plików.

```
#!/bin/sh

COUNTER=0

if [ "$#" != "1" ]; then
    echo "Skrypt uruchamiany z parametrem (sciezka do katalogu)"
    exit 1
fi

if [ ! -d $1 ]; then
    echo "Nie ma takiego katalogu"
    exit 1
fi

for file in `ls -r -S "$1/"`; do
    if [ -x "$1/$file" ]; then
        mv "$1/$file" "$1/$file.$COUNTER"
        COUNTER=$((COUNTER+1))
    fi
done
```

Skrypt nr 5

Łączenie w jeden wszystkich plików należących do zadanego katalogu (parametr wywołania skryptu), o zadanym rozszerzeniu (drugi parametr skryptu). Kolejność, w jakiej pliki zostaną połączone - nieistotna. Treść każdego pliku powinna być poprzedzona nagłówkiem z jego nazwą.

```
#!/bin/sh

NOWY="nowy"

if [ "$#" != "2" ]; then
    echo "Skrypt uruchamiany z 2 parametrami"
    exit 1
fi

if [ ! -d $1 ]; then
    echo "Nie ma takiego katalogu"
    exit 2
fi

for file in `ls $1/*. $2`; do
    echo "+++++$file++++" >> $1/$NOWY.$2
    cat $file >> $1/$NOWY.$2
done
```

Skrypt nr 6

Łączenie w jeden plików z listy, umieszczonej w pliku o zadanej nazwie (parametr wywołania skryptu). Kolejność, w jakiej pliki zostaną połączone - ściśle według listy. Treść każdego pliku powinna być poprzedzona nagłówkiem z jego nazwą. Plik wynikowy powinien mieć nazwę pliku pierwotnie zawierającego listę.

```
#!/bin/sh

if [ "$#" != "2" ]; then
    echo "Skrypt uruchamiany z 2 parametrami"
    exit 1
fi

if [ ! -d $1 ]; then
    echo "Nie ma takiego katalogu"
    exit 2
fi

if [ ! -r $2 ]; then
    echo "Nie mozna czytac z pliku"
    exit 3
fi

for file in `cat $2`; do
    echo "----$file----" >> $1/$2
    cat $1/$file >> $1/$2
done
```

Skrypt nr 7

Przesuwanie wszystkich plików z ustawionym prawem wykonywania z jednego katalogu do drugiego. Pozostałe pliki w katalogu nie powinny być ruszane. Nazwy katalogów, źródłowego i docelowego, zadawane jako parametry skryptu.

```
#!/bin/sh

if [ "$#" != "2" ]; then
    echo "Skrypt uruchamiany z 2 parametrami"
    exit 1
fi

if [ ! -d $1 ] || [ ! -d $2 ]; then
    echo "Nie ma takiego katalogu"
    exit 2
fi

for file in `ls $1/`; do
    if [ -x $1/$file ]; then
        mv $1/$file $2/$file
    fi
done
```

Skrypt nr 8

Listowanie plików z zadanego katalogu (parametr wywołania skryptu), wraz z jego podkatalogami. Zawartość podkatalogów ma być listowana w postaci jściezka dostępu względem listowanego katalogu i nazwa pliku. Ograniczyć liczbę zagnieżdżeń podkatalogów (np. do dwóch).

```
#!/bin/sh

MAX=2
ilosc=0

listowanie () {
    local katalog=$1
    local ilosc=$((ilosc+1))
    for file in `ls $katalog`; do
        echo "$katalog/$file"
        if [ -d "$katalog/$file" ] && [ "$MAX" -ge "$ilosc" ]; then listowanie "$katalog/$file"; fi
    done;
}

if [ "$#" != "1" ]; then
    echo "Skrypt uruchamiany z 1 parametrem"
    exit 1
fi

if [ ! -d $1 ]; then
    echo "Nie ma takiego katalogu"
    exit 2
fi

listowanie $1
```

Skrypt nr 9

Kasowanie wszystkich plików pustych (o zerowej wielkości) w zadanym katalogu (parametr wywołania skryptu). Skrypt powinien tworzyć w zadanym pliku listę skasowanych plików. Nie powinien analizować dołączy symbolicznych.

```
#!/bin/sh

NOWY="nowy.txt"

if [ "$#" != "1" ]; then
    echo "Skrypt uruchamiany z 1 parametrem"
    exit 1
fi

if [ ! -d $1 ]; then
    echo "Nie ma takiego katalogu"
    exit 2
fi

for file in `ls $1/`; do
    if [ ! -s $1/$file ] && [ ! -L $1/$file ]; then
        echo "$file" >> $1/$NOWY
        rm -rf $1/$file
    fi
done
```

Skrypt nr 10

Porównanie zawartości dwóch zadanych katalogów (argumenty skryptu). Przy porównaniu należy ignorować podkatalogi. W wyniku wyświetlić na ekranie listę plików o identycznych nazwach w obu katalogach.

```
#!/bin/sh

if [ "$#" != 2 ]; then
    echo "Skrypt uruchamiany z dwoma argumentami"
    exit 1
fi

if [ ! -d $1 ] || [ ! -d $2 ]; then
    echo "Nie znaleziono katalogu docelowego"
    exit 2
fi

for file in `ls $1`; do
    if [ ! -d $1/$file ] && [ -e $2/$file ]; then
        echo "$file"
    fi
done
```

Skrypt nr 11

Porównanie zawartości zadanego katalogu z listą plików (nazwa katalogu i pliku z listą zadawana w argumentach skryptu). Należy wygenerować listę plików brakujących w katalogu i takich, które nie są na liście.

```
#!/bin/sh

if [ "$#" != 2 ]; then
    echo "Skrypt uruchamiany z dwoma argumentami"
    exit 1
fi

if [ ! -d $1 ]; then
    echo "Nie znaleziono katalogu docelowego"
    exit 2
fi

if [ ! -r $2 ]; then
    echo "Nie znaleziono pliku z lista"
    exit 3
fi

echo "---tych plikow nie ma w katalogu---" >> "$1/nowy.txt"
for file in `cat $2`; do
    if [ ! -e $1/$file ]; then
        echo $file >> "$1/nowy.txt"
    fi
done

echo "---tych plikow nie ma na liscie---" >> "$1/nowy.txt"
for file in `ls $1`; do
    zmienna=false
    for plik in `cat $2`; do
        if [ $file = $plik ]; then
            zmienna=true
        fi
    done
    if [ $zmienna = false ]; then
        echo $file >> "$1/nowy.txt"
    fi
done
```

Skrypt nr 12

Usuwanie wszystkich podkatalogów zadanego katalogu (parametr wywołania skryptu). Zawartość tych podkatalogów należy przenieść do katalogu nadrzędnego. Usuwanie dotyczy tylko jednego poziomu podkatalogów.

```
#!/bin/sh

if [ "$#" != "1" ]; then
    echo "Skrypt uruchamiany z parametrem"
    exit 1
fi

if [ ! -d $1 ]; then
    echo "Nie ma takiego katalogu"
    exit 2
fi

for file in `ls $1`; do
    if [ -d $1/$file ]; then
        mv $1/$file/* $1/
        rm -rf $1/$file
    fi
done
```

Skrypt nr 13

Usuwanie z zadanego katalogu (pierwszy argument skryptu) wszystkich plików, których nazwy pokrywają się z nazwami w drugim katalogu (drugi argument). Nie usuwaj podkatalogów. Nie usuwaj również plików o nazwach pokrywających się z nazwami podkatalogów w drugim katalogu.

```
#!/bin/sh

if [ "$#" != 2 ]; then
    echo "Skrypt uruchamiany z dwoma argumentami"
    exit 1
fi

if [ ! -d $1 ] || [ ! -d $2 ]; then
    echo "Nie znaleziono katalogu docelowego"
    exit 2
fi

for file in `ls $1`; do
    if [ ! -d $1/$file ]; then
        if [ -e $2/$file ] && [ ! -d $2/$file ]; then
            rm -rf $1/$file
        fi
    fi
done
```

Skrypt nr 14

Ustawienie na aktualny czas ostatniej modyfikacji wszystkich plików zadanego katalogu (parametr wywołania skryptu), do których mamy prawo dostępu do zapisu/modyfikacji.

```
#!/bin/sh

if [ "$#" != "1" ]; then
    echo "Skrypt uruchamiany z parametrem"
    exit 1
fi

if [ ! -d $1 ]; then
    echo "Nie ma takiego katalogu"
    exit 2
fi

for file in `ls $1`; do
    if [ -w $1/$file ]; then
        touch "$1/$file"
    fi
done
```

Skrypt nr 15

Policzenie wszystkich plików w zadanym katalogu (parametr wywołania skryptu), do których ustawione jest prawo dostępu do wykonania („execute”).

```
#!/bin/sh

COUNTER=0

if [ "$#" != "1" ]; then
    echo "Skrypt uruchamiany z parametrem"
    exit 1
fi

if [ ! -d $1 ]; then
    echo "Nie ma takiego katalogu"
    exit 2
fi

for file in `ls $1`; do
    if [ -x $1/$file ]; then
        COUNTER=$((COUNTER+1))
    fi
done

echo "Liczba plikow wykonywalnych: $COUNTER"
```

Skrypt nr 16

Tworzenie nowych plików w zadanym katalogu (parametr wywołania skryptu), według listy umieszczonej w pliku (drugi parametr wywołania skryptu). Nowe pliki mają być zerowej wielkości (puste). Jeżeli jakiś plik już istnieje, to nie powinien zostać zniszczony. Na liście mogą znajdować się ścieżki względne do plików, w takim wypadku należy stworzyć brakujące katalogi.

```
#!/bin/sh

if [ "$#" != "2" ]; then
    echo "Skrypt uruchamiany z dwoma parametrami"
    exit 1
fi

if [ ! -d $1 ] || [ ! -w "$1" ]; then
    echo "Nie ma takiego katalogu"
    exit 2
fi

if [ ! -r $2 ]; then
    echo "Nie można czytać z pliku lub plik nie istnieje"
    exit 3
fi

for file in `cat $2`; do
    if [ ! -e "$1/$file" ]; then
        mkdir -p $1/`dirname $file`
        touch $1/$file
    fi
done
```

Skrypt nr 17

Jest to modyfikacja skryptu nr 12. Polega jedynie na tym, że skrypt wykrywa kolizję nazw plików w katalogu, do którego wrzucane są dane i w razie wystąpienia powtórzeń dodaje końcówki "*.i", gdzie i jest numerem kolejnego powtórzenia.

```
#!/bin/sh

COUNTER=0

zamien () {
    for czek in `ls $1`; do
        if [ $2.$COUNTER = $czek ]; then
            COUNTER=$((COUNTER+1))
            zamien $1 $2
        fi
    done
}

if [ "$#" != "1" ]; then
    echo "Skrypt uruchamiany z parametrem"
    exit 1
fi

if [ ! -d $1 ]; then
    echo "Nie ma takiego katalogu"
    exit 2
fi

for file in `ls $1`; do
    if [ -d $1/$file ]; then
        for plik in `ls $1/$file`; do
            if [ -e "$1/$plik" ]; then
                COUNTER=0
                zamien $1 $plik
                mv -f $1/$file/$plik $1/$plik.$COUNTER
            else
                mv -f $1/$file/$plik $1
            fi
        done
    fi
done
rm -rf $1/$file
```

Skrypt nr 18

W zadanym katalogu (parametr wywołania skryptu) zamienić wszystkie spacje w nazwach plików na podkreślenia. Wielokrotne spacje zamienić w pojedyncze podkreślenie.

```
#!/bin/sh

if [ "$#" != "1" ]; then
    echo "Skrypt uruchamiany z 1 parametrem"
    exit 1
fi

if [ ! -d $1 ]; then
    echo "Nie ma takiego katalogu"
    exit 2
fi

for file in $1/*;do
    nowy='echo "$file" | tr -s ' ' | tr ' ' '_' '
    if [ ! -d "$1/$file" ] && [ ! "$file" = $nowy ] ; then
        mv "$file" $nowy
    fi
done
```

Skrypt nr 19

W zadanym pliku (parametr wywołania skryptu) w poszczególnych liniach znajduje się ścieżka i litera d lub f (katalog lub plik). Ścieżka wskazuje gdzie a litera co utworzyć.

```
#!/bin/sh

if [ "$#" != "2" ]; then
    echo "Skrypt uruchamiany z dwoma parametrami"
    exit 1
fi

if [ ! -d $1 ] || [ ! -w $1 ]; then
    echo "Nie ma takiego katalogu"
    exit 2
fi

if [ ! -r $2 ]; then
    echo "Nie można czytać z pliku lub plik nie istnieje"
    exit 3
fi

bit=0
name=""
dir="d"
fil="f"
for file in `cat $2`; do
    if [ $bit = 0 ]; then
        name=$file
        mkdir -p `dirname $file`
        bit=1
    elif [ $bit = 1 ]; then
        if [ $file = $dir ]; then
            mkdir $name
        elif [ $file = $fil ]; then
            touch $name
        fi
        bit=0
    fi
done
```

Skrypt nr 20

W zadanym katalogu znaleźć pliki identyczne, które nie są dowiązaniami twardymi do siebie i zrobić z nich dowiązania twarde.

```
if [ "$#" != "1" ]; then
    echo "Skrypt uruchamiany z jednym parametrem"
    exit 1
fi

if [ ! -d $1 ]; then
    echo "Podany katalog nie istnieje"
    exit 2
fi

for plik1 in `ls $1`; do
    if [ -f $1/$plik1 ]; then
        for plik2 in `ls $1`; do
            inode1=`ls -li $1/$plik1 | cut -f1 -d' '`
            inode2=`ls -li $1/$plik2 | cut -f1 -d' '`
            if [ "$plik1" != "$plik2" ] && [ -f $1/$plik2 ]; then
                cmp -s $1/$plik1 $1/$plik2 #porównanie zawartości dwóch plików
                if [ $? -eq 0 ] && [ $inode1 != $inode2 ]; then
```

```
        rm -f $1/$plik2
        ln $1/$plik1 $1/$plik2
    fi
done
fi
done
```