

## Ćwiczenie 3

### Najprostsze skrypty w bash-u

#### Cel ćwiczenia:

Przedmiotem ćwiczenia jest przedstawienie możliwości napisania prostych skryptów w środowisku bash-a.

#### Przebieg ćwiczenia - zagadnienia:

1. Napisz skrypt jak poniżej:
  - a. treść

```
psm@karas: ~  
# Napisz skrypt, który wyświetla w kolejnych liniach wartości  
# od wartości zadanej jako argument wywołania tego skryptu  
# do wartości 1 z krokiem -1  
# Skrypt powinien zawierać sprawdzanie obecności argumentu wywołania  
# tego skryptu.  
if [ $# -ne 1 ] ; then  
    echo "Skrypt wymaga parametru - liczby całkowitej dodatniej."  
else  
    i=$1  
    while [ $i -gt 0 ] ; do  
        echo $i  
        i=`expr $i - 1`  
    done  
fi  
psm@karas:~$
```

- b. wykonanie

```
psm@karas: ~  
psm@karas:~$ ./skrypt1  
Skrypt wymaga parametru - liczby całkowitej dodatniej.  
psm@karas:~$ ./skrypt1 3  
3  
2  
1  
psm@karas:~$ ./skrypt1 5  
5  
4  
3  
2  
1  
psm@karas:~$
```

2. Napisz skrypt jak poniżej
  - a. treść

```
psm@karas: ~  
psm@karas:~$ cat skrypt2  
# Napisz skrypt wyświetlający wszystkie pliki z rozszerzeniem .doc oraz .txt  
# znajdujące się w bieżącym katalogu, wypisując następujący komunikat:  
# Plik o nazwie <tu nazwa pliku> jest plikiem tekstowym.  
for z in *.doc *.txt ; do  
    echo "Plik o nazwie $z jest plikiem tekstowym."  
done  
psm@karas:~$
```

b. dane

```
psm@karas: ~  
psm@karas:~$ ls -l  
total 0  
-rw-r--r-- 1 psm psm  0 Apr  2 14:15 a1.doc  
-rw-r--r-- 1 psm psm  0 Apr  2 14:15 a1.txt  
-rw-r--r-- 1 psm psm  0 Apr  2 14:15 a1.xls  
-rw-r--r-- 1 psm psm  0 Apr  2 14:15 a2.doc  
-rw-r--r-- 1 psm psm  0 Apr  2 14:15 a2.xls  
-rw-r--r-- 1 psm psm  0 Apr  2 14:15 c.exe  
drwxr-xr-x 1 psm psm 512 Apr  2 14:20 plik.txt  
-rwxrwxrwx 1 psm psm 295 Apr  2 14:19 skrypt2  
psm@karas:~$
```

c. wynik działania

```
psm@karas: ~  
psm@karas:~$ ./skrypt2  
Plik o nazwie a1.doc jest plikiem tekstowym.  
Plik o nazwie a2.doc jest plikiem tekstowym.  
Plik o nazwie a1.txt jest plikiem tekstowym.  
Plik o nazwie plik.txt jest plikiem tekstowym.  
psm@karas:~$
```

3. Napisz skrypt jak poniżej

a. treść

```
psm@karas: ~  
psm@karas:~$ cat skrypt3  
# Napisz skrypt wypisujący odpowiedni komunikat na temat bieżącej daty w języku  
# zależnym od argumentu wywołania tego skryptu:  
# dla a - data w języku angielskim  
# dla p - data w języku polskim  
# Komunikaty powinny mieć odpowiednio następującą postać:  
# Today is Friday, 2 of April, 2021  
# Dzisiaj jest piątek, 2 kwietnia 2021 roku.  
# Skrypt powinien być wyposażony w sprawdzenie wywołania tego skryptu  
# z dokładnie jednym argumentem.  
if test $# != 1 ; then  
    echo "Program wymaga podania dokładnie jednego argumentu wywołania:"  
    echo "  a - data w wersji angielskiej"  
    echo "  p - data w wersji polskiej"  
else  
    case $1 in  
        a) echo "data w wersji angielskiej" ;;  
        p) set `date`  
            case $1 in  
                Mo*) DZIEN=poniedziałek ;;  
                Tu*) DZIEN=wtorek ;;  
                We*) DZIEN=środa ;;  
                Th*) DZIEN=czwartek ;;  
                Fr*) DZIEN=piątek ;;  
                Sa*) DZIEN=sobota ;;  
                Su*) DZIEN=niedziela ;;  
            esac  
            case $2 in  
                Ja*) MIESIAC=stycznia ;;  
                Fe*) MIESIAC=lutego ;;  
                Ma*) MIESIAC=marca ;;  
                Ap*) MIESIAC=kwietnia ;;  
                Ma*) MIESIAC=maja ;;  
            esac  
            echo Dzisiaj jest $DZIEN, $3 $MIESIAC $6 roku.  
    esac  
fi  
psm@karas:~$
```

b. wynik działania

```
psm@karas: ~  
psm@karas:~$ ./skrypt3 p  
Dzisiaj jest piątek, 2 kwietnia 2021 roku.  
psm@karas:~$
```

4. Napisz skrypt jak poniżej

a. treść

```
psm@karas: ~  
psm@karas:~$ cat skrypt4  
# Napisz skrypt, który wykonuje następujące operacje na zadanym pliku wejściowym:  
# 1. linie, które zawierają parzystą liczbę słów zostaną zapisane w pliku  
#   o nazwie PARZYSTE  
# 2. linie, które zawierają nieparzystą liczbę słów zostaną zapisane w pliku  
#   o nazwie NIEPARZYSTE  
# 3. zostanie wypisany komunikat mówiący o liczbie linii z parzystą  
#   i nieparzystą liczbą słów.  
# Nazwa pliku jest podawana jako argument wywołania tego skryptu.  
# Skrypt powinien zawierać mechanizmy sprawdzania poprawności uruchomienia  
# tego skryptu - liczbę argumentów - oraz istnienia pliku wejściowego.  
if [ ! $# -eq 1 ] ; then  
    echo "Skrypt powinien być wywołany z jednym argumentem - nazwą pliku wejściowego."  
else  
    PLIKWE=$1  
    LPAR=0  
    LNP=0  
    if [ ! -f $PLIKWE ] ; then  
        echo "Plik wejściowy o nazwie $PLIKWE nie istnieje."  
    else  
        while read LINIA ; do  
            WYNIK=$((`echo $LINIA | wc -w` % 2))  
            if [ $WYNIK -eq 0 ] ; then  
                echo $LINIA >> PARZYSTE  
                let LPAR++  
            else  
                echo $LINIA >> NIEPARZYSTE  
                let LNP++  
            fi  
        done < "$PLIKWE"  
        echo "W pliku wejściowym o nazwie $PLIKWE było $LPAR linii z parzystą i $LNP linii z nieparzystą liczbą słów."  
    fi  
fi  
psm@karas:~$
```

b. plik z danymi

```
psm@karas: ~  
psm@karas:~$ cat wejscie  
Ala ma kota  
A Ola ma psa  
Dzisiaj jest ładna pogoda  
Jutro będzie deszcz  
Też  
psm@karas:~$
```

c. efekt działania programu

```
psm@karas: ~  
psm@karas:~$ ./skrypt4 wejscie  
W pliku wejściowym o nazwie wejscie było 2 linii z parzystą i 3 linii z nieparzystą liczbą słów.  
psm@karas:~$ cat PARZYSTE  
A Ola ma psa  
Dzisiaj jest ładna pogoda  
psm@karas:~$ cat NIEPARZYSTE  
Ala ma kota  
Jutro będzie deszcz  
Też  
psm@karas:~$
```

d. jak widać dane zostały przetworzone prawidłowo.

5. Napisz skrypt jak poniżej:

a. treść

```
psm@karas: ~  
psm@karas:~$ cat skrypt1  
# Napisz skrypt, który dla każdego pliku, znajdującego się w bieżącym katalogu,  
# o nazwie spełniającej następujące warunki:  
# 1. rozpoczynającej się od dowolnego znaku,  
# 2. na kolejnych znakach posiadającej string 2017  
# 3. i dalej mogących mieć dowolne znaki w nazwie pliku,  
# 4. posiadający rozszerzenie nazwy txt  
# określi ilość zawartych w nim poszczególnych liczb całkowitych: parzystych,  
# nieparzystych i zero. Zakładamy, że dane zawarte w plikach są poprawne - tzn.  
# pliki nie zawierają innych danych niż wyżej wymienione. Przykładowy komunikat  
# będący efektem działania tego skryptu przedstawiono poniżej:  
# Plik a201701.txt zawiera: 10 parzystych i 5 nieparzystych liczb oraz 5 liczb zero.
```

b. dane - pliki wejściowe i zawartość wybranego pliku:

```
psm@karas: ~  
psm@karas:~$ ls -l  
total 8  
drwxr-xr-x 1 psm psm 512 Apr 9 14:48 ALFA  
drwxr-xr-x 1 psm psm 512 Apr 9 14:49 BETA  
drwxr-xr-x 1 psm psm 512 Apr 9 14:48 GAMMA  
-rw-r--r-- 1 psm psm 39 Apr 9 14:12 a201601.txt  
-rw-r--r-- 1 psm psm 60 Apr 9 14:11 a201701.txt  
drwxr-xr-x 1 psm psm 512 Apr 9 14:12 b201702.txt  
-rwxr-xr-x 1 psm psm 1247 Apr 9 15:29 skrypt1  
-rwxr-xr-x 1 psm psm 626 Apr 9 15:00 skrypt2  
psm@karas:~$ cat a201701.txt  
2 4 6 8 11 33 55 0 0  
7777 99999 0 0 0  
46 58 88 100 200 1024  
psm@karas:~$
```

c. treść skryptu:

```
psm@karas: ~  
psm@karas:~$ cat skrypt1  
# Napisz skrypt, który dla każdego pliku, znajdującego się w bieżącym katalogu,  
# o nazwie spełniającej następujące warunki:  
# 1. rozpoczynającej się od dowolnego znaku,  
# 2. na kolejnych znakach posiadającej string 2017  
# 3. i dalej mogących mieć dowolne znaki w nazwie pliku,  
# 4. posiadający rozszerzenie nazwy txt  
# określi ilość zawartych w nim poszczególnych liczb całkowitych: parzystych,  
# nieparzystych i zero. Zakładamy, że dane zawarte w plikach są poprawne - tzn.  
# pliki nie zawierają innych danych niż wyżej wymienione. Przykładowy komunikat  
# będący efektem działania tego skryptu przedstawiono poniżej:  
# Plik a201701.txt zawiera: 10 parzystych i 5 nieparzystych liczb oraz 5 liczb ze  
ro.  
for PLIK in `ls ?2017*.txt` ; do  
  if [ -f $PLIK ] ; then  
    LP=0  
    LNP=0  
    LZ=0  
  
    while read LINIA ; do  
      for LICZBA in $LINIA ; do  
        if [ $LICZBA -eq 0 ] ; then  
  
          let LZ++  
        else  
          WYNIK=$(( $LICZBA % 2 ))  
          if [ $WYNIK -eq 0 ] ; then  
            let LP++  
          else  
            let LNP++  
          fi  
        fi  
      done  
    done < "$PLIK"  
    echo "Plik $PLIK zawiera $LP parzystych i $LNP nieparzystych liczb"  
    echo "oraz $LZ liczb zero."  
  fi  
done  
psm@karas:~$
```

d. wykonanie

```
psm@karas: ~  
psm@karas:~$ ./skrypt1  
Plik a201701.txt zawiera 10 parzystych i 5 nieparzystych liczb  
oraz 5 liczb zero.  
psm@karas:~$ cat a201701.txt  
2 4 6 8 11 33 55 0 0  
7777 99999 0 0 0  
46 58 88 100 200 1024  
psm@karas:~$
```

6. Napisz skrypt jak poniżej

a. treść

```
psm@karas: ~  
psm@karas:~$ cat skrypt2  
# Napisz skrypt, który wypisze nazwy tych katalogów z katalogu bieżącego,  
# które zawierają jednocześnie: co najmniej dwa pliki do odczytu i co najmniej  
# trzy pliki wykonywalne.
```

b. dane

```
psm@karas: ~  
psm@karas:~$ ls -l  
total 4  
drwxr-xr-x 1 psm psm 512 Apr  9 14:48 ALFA  
drwxr-xr-x 1 psm psm 512 Apr  9 14:49 BETA  
drwxr-xr-x 1 psm psm 512 Apr  9 14:48 GAMMA  
-rwxr-xr-x 1 psm psm 626 Apr  9 15:00 skrypt2  
psm@karas:~$ cd ALFA  
psm@karas:~/ALFA$ ls -l  
total 0  
--wx--x--x 1 psm psm 0 Apr  9 14:48 a1  
--w----- 1 psm psm 0 Apr  9 14:48 a2  
--wx--x--x 1 psm psm 0 Apr  9 14:48 a3  
--w----- 1 psm psm 0 Apr  9 14:48 a4  
-rwxr-xr-x 1 psm psm 0 Apr  9 14:48 a5  
-rw-r--r-- 1 psm psm 0 Apr  9 14:48 a6  
psm@karas:~/ALFA$ cd ..  
psm@karas:~$ cd BETA  
psm@karas:~/BETA$ ls -l  
total 0  
--wx--x--x 1 psm psm 0 Apr  9 14:49 b1  
--wx--x--x 1 psm psm 0 Apr  9 14:49 b2  
-rwxr-xr-x 1 psm psm 0 Apr  9 14:49 b3  
--wx--x--x 1 psm psm 0 Apr  9 14:49 b4  
--w----- 1 psm psm 0 Apr  9 14:49 b5  
psm@karas:~/BETA$ cd ..  
psm@karas:~$ cd GAMMA  
psm@karas:~/GAMMA$ ls -l  
total 0  
psm@karas:~/GAMMA$ cd ..  
psm@karas:~$
```

c. treść skryptu:

```
psm@karas: ~  
# Napisz skrypt, który wypisze nazwy tych katalogów z katalogu bieżącego,  
# które zawierają jednocześnie: co najmniej dwa pliki do odczytu i co najmniej  
# trzy pliki wykonywalne.  
for KATALOG in * ; do  
  if [ -d $KATALOG ] ; then  
    cd $KATALOG  
    LR=0  
    LX=0  
    for PLIK in * ; do  
      if [ -f $PLIK ] ; then  
        if [ -x $PLIK ] ; then  
          let LX++  
        fi  
        if [ -r $PLIK ] ; then  
          let LR++  
        fi  
      fi  
    done  
    if [ $LR -gt 1 -a $LX -gt 2 ] ; then  
      echo "Katalog $KATALOG zawiera minimum dwa pliki do odczytu i trzy pliki wykonywalne."  
    fi  
    cd ".."  
  fi  
done  
psm@karas:~$
```

d. wynik działania

```
psm@karas: ~  
psm@karas:~$ ./skrypt2  
Katalog ALFA zawiera minimum dwa pliki do odczytu i trzy pliki wykonywalne.  
psm@karas:~$
```

7. Napisz skrypt jak poniżej

a. treść

```
psm@karas: ~  
psm@karas:~$ cat skrypt3  
# Napisz skrypt, który wykona następujące operacje na pliku wejściowym:  
# 1. do każdej linii pliku doda na jej początku tekst w postaci stringu "linia"  
# wraz z bieżącym numerem linii (np. linia 45)  
# 2. jeżeli linia nie zawiera żadnych znaków, to w linii tej należy napisać  
# komunikat o następującej postaci: "# << TA LINIA INTENCJONALNIE JEST PUSTA ! >>"  
# 3. napisz na końcu pliku informacje mówiące o liczbie pustych i niepustych liniach  
# pliku oraz liczbę znaków i słów zawartych w tym pliku.  
# Nazwa pliku wejściowego jest podawana jako argument wywołania tego skryptu.  
# Skrypt powinien zawierać mechanizmy sprawdzania poprawności uruchomienia tego skryptu:  
# liczbą argumentów (jeden) oraz istnienie pliku wejściowego.
```

b. plik wejściowy o nazwie plikwejsciowy

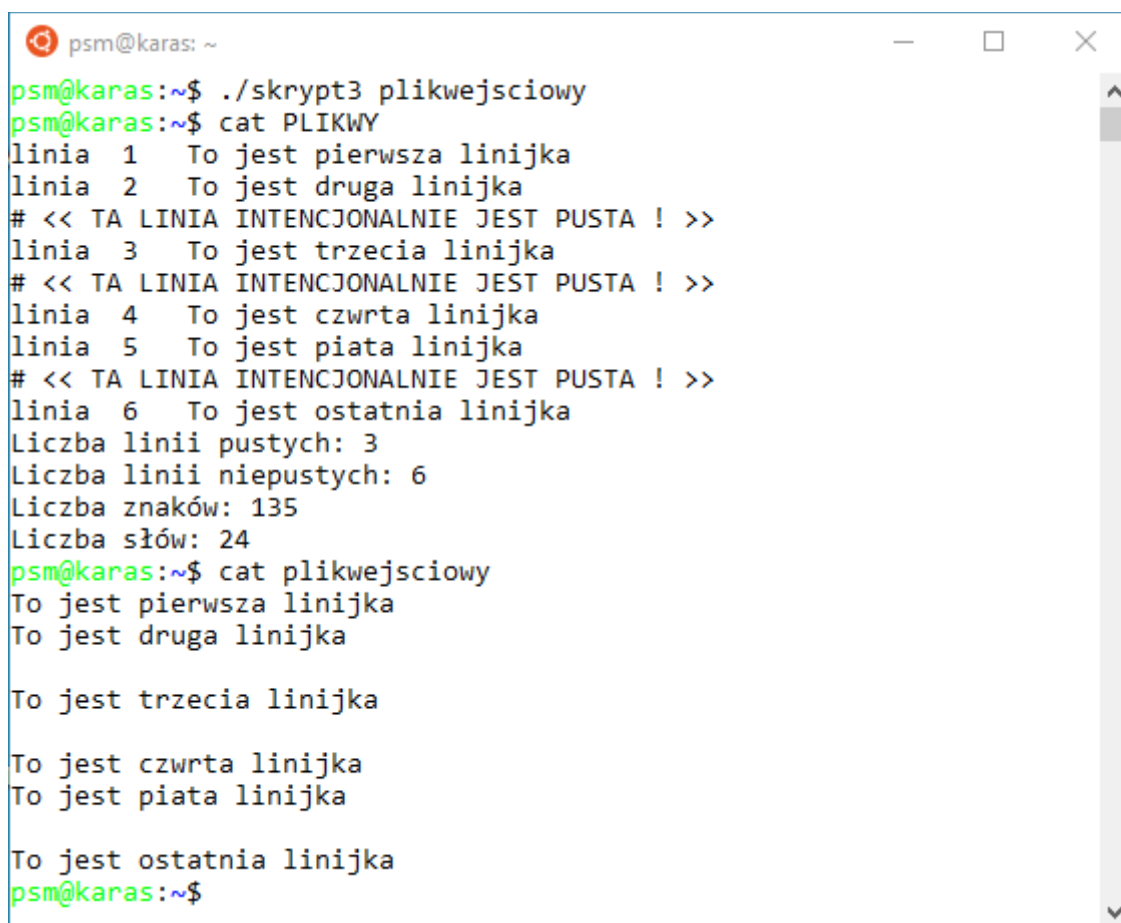


```
psm@karas: ~  
psm@karas:~$ cat plikwejsciowy  
To jest pierwsza linijka  
To jest druga linijka  
  
To jest trzecia linijka  
  
To jest czwarta linijka  
To jest piata linijka  
  
To jest ostatnia linijka  
psm@karas:~$
```

c. treść skryptu

```
psm@karas: ~  
psm@karas:~$ cat skrypt3  
# Napisz skrypt, który wykona następujące operacje na pliku wejściowym:  
# 1. do każdej linii pliku doda na jej początku tekst w postaci stringu "linia"  
# wraz z bieżącym numerem linii (np. linia 45)  
# 2. jeżeli linia nie zawiera żadnych znaków, to w linii tej należy napisać  
# komunikat o następującej postaci: "# << TA LINIA INTENCJONALNIE JEST PUSTA ! >>"  
# 3. napisz na końcu pliku informacje mówiące o liczbie pustych i niepustych liniach  
# pliku oraz liczbę znaków i słów zawartych w tym pliku.  
# Nazwa pliku wejściowego jest podawana jako argument wywołania tego skryptu.  
# Skrypt powinien zawierać mechanizmy sprawdzania poprawności uruchomienia tego skryptu:  
# liczbą argumentów (jeden) oraz istnienie pliku wejściowego.  
if [ $# -ne 1 ] ; then  
    echo "Skrypt wymaga jednego argumentu - nazwy pliku wejściowego."  
else  
    PLIKWE=$1  
    if [ ! -f $PLIKWE ] ; then  
        echo "Plik o podanej nazwie $PLIKWE nie istnieje w bieżącym katalogu."  
    else  
        NRLINII=1  
        LLNP=0  
        LLP=0  
        LZ=0  
        LS=0  
        while read LINIA ; do  
            WYNIK=$((`echo $LINIA | wc -c` - 1 ))  
            if [ $WYNIK -eq 0 ] ; then  
                echo "# << TA LINIA INTENCJONALNIE JEST PUSTA ! >>" >> PLIKWY  
                let LLP++  
            else  
                echo "linia " $NRLINII " " $LINIA >> PLIKWY  
                let LLNP++  
                let NRLINII++  
                LZNAKOW=$((`echo $LINIA | wc -c` - 1))  
                LSLOW=$((`echo $LINIA | wc -w`))  
                LZ=`expr $LZ + $LZNAKOW`  
                LS=`expr $LS + $LSLOW`  
            fi  
        done < "$PLIKWE"  
        echo "Liczba linii pustych: $LLP" >> PLIKWY  
        echo "Liczba linii niepustych: $LLNP" >> PLIKWY  
        echo "Liczba znaków: $LZ" >> PLIKWY  
        echo "Liczba słów: $LS" >> PLIKWY  
    fi  
fi  
# cp PLIKWY $PLIKWE  
psm@karas:~$
```

d. wynik działania – porównanie z plikiem wejściowym:



```
psm@karas: ~  
psm@karas:~$ ./skrypt3 plikwejsciowy  
psm@karas:~$ cat PLIKWY  
linia 1   To jest pierwsza linijka  
linia 2   To jest druga linijka  
# << TA LINIA INTENCJONALNIE JEST PUSTA ! >>  
linia 3   To jest trzecia linijka  
# << TA LINIA INTENCJONALNIE JEST PUSTA ! >>  
linia 4   To jest czwarta linijka  
linia 5   To jest piata linijka  
# << TA LINIA INTENCJONALNIE JEST PUSTA ! >>  
linia 6   To jest ostatnia linijka  
Liczba linii pustych: 3  
Liczba linii niepustych: 6  
Liczba znaków: 135  
Liczba słów: 24  
psm@karas:~$ cat plikwejsciowy  
To jest pierwsza linijka  
To jest druga linijka  
  
To jest trzecia linijka  
  
To jest czwarta linijka  
To jest piata linijka  
  
To jest ostatnia linijka  
psm@karas:~$
```