
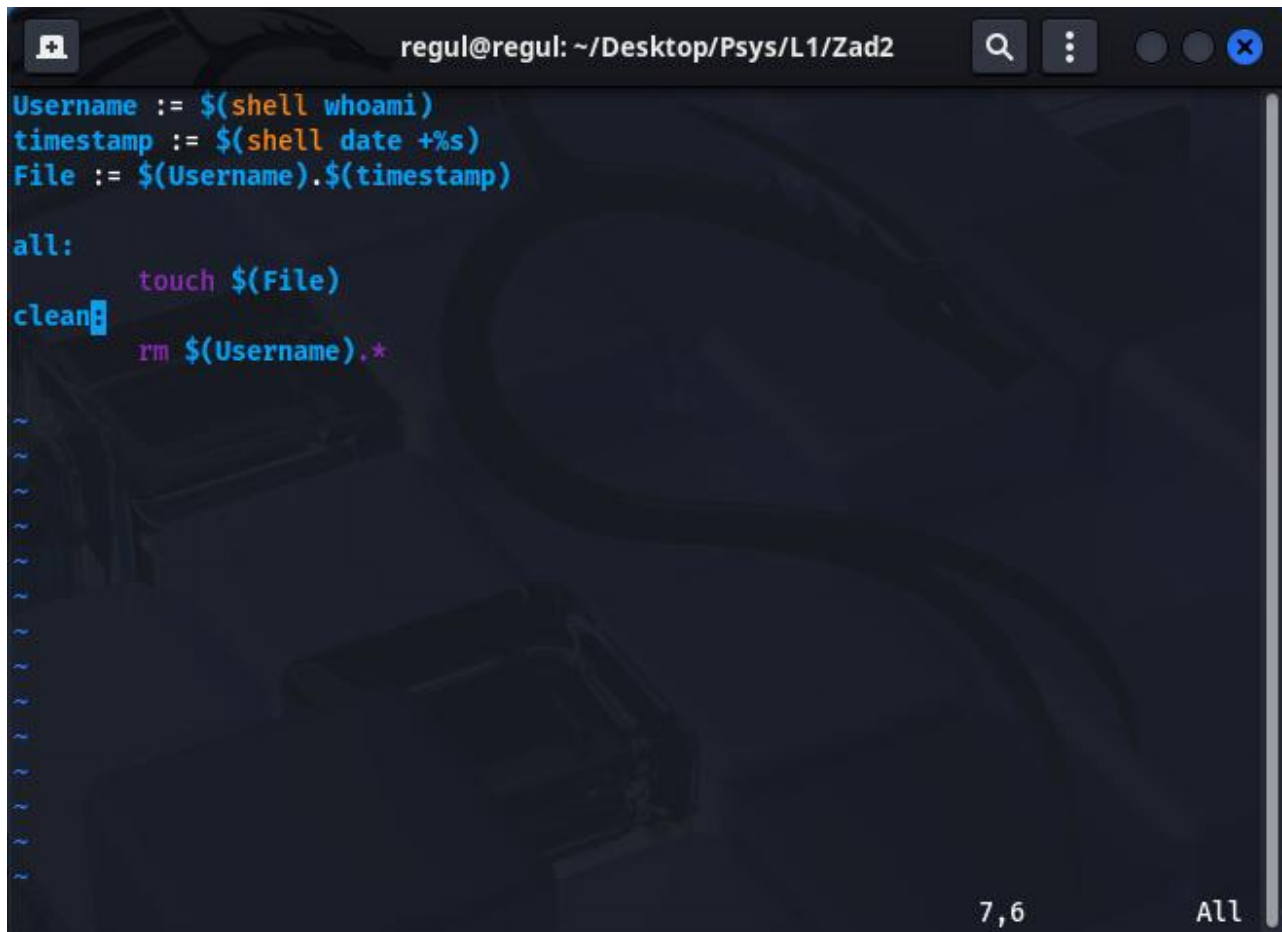


<p>POLITECHNIKA WROCŁAWSKA</p>  <p>Wydział Informatyki i Telekomunikacji</p>	<p>Wydział: Informatyki i Telekomunikacji</p> <p>Kierunek: Cyberbezpieczeństwo</p> <p>Rok Akademicki: 2024/2025</p> <p>Rok studiów, semestr: 3</p> <p>Grupa: 6</p> <p>Termin: <i>środa, 15:15</i></p>
<p>Programowanie systemowa</p>	
<p>Prowadzący:</p> <p>mgr inż. Przemysław Świercz</p>	<p>Autor: Jakub Regulski, 283866</p>
<p>Data wykonania ćwiczenia:</p> <p>22.10.2025</p>	
<p>Data oddania sprawozdania:</p> <p>27.10.2025</p>	

Zadanie 1



```
regul@regul: ~/Desktop/Phys/L1/Zad2
Username := $(shell whoami)
timestamp := $(shell date +%s)
File := $(Username).$(timestamp)

all:
    touch $(File)
clean:
    rm $(Username)*
```

Makefile z pierwszego zadania

W pierwszej linii do zmiennej „Username” przypisujemy nazwę aktualnie zalogowanego użytkownika, do zmiennej „timestamp” przypisujemy aktualną liczbę sekund która minęła od 1 stycznia 1970, zmienna File przechowuje nazwę pliku. Makefile zawiera 2 targety. Target „all” służy do utworzenia pliku, natomiast „clean” usuwa nam wszystkie utworzone wcześniej pliki.

```
(regul@regul)-[~/Desktop/Psys/L1/Zad2]
$ make
touch regul.1761572429

(regul@regul)-[~/Desktop/Psys/L1/Zad2]
$ ls
Makefile  regul.1761572429

(regul@regul)-[~/Desktop/Psys/L1/Zad2]
$ make clean
rm regul.*

(regul@regul)-[~/Desktop/Psys/L1/Zad2]
$ ls
Makefile
```

wykoanie make oraz make clean

Zadanie 2

```
regul@regul: ~/Desktop/Psys/L1/Zad3
Username := $(shell whoami)
timestamp := $(shell date +%s)
File := $(Username).$(timestamp)

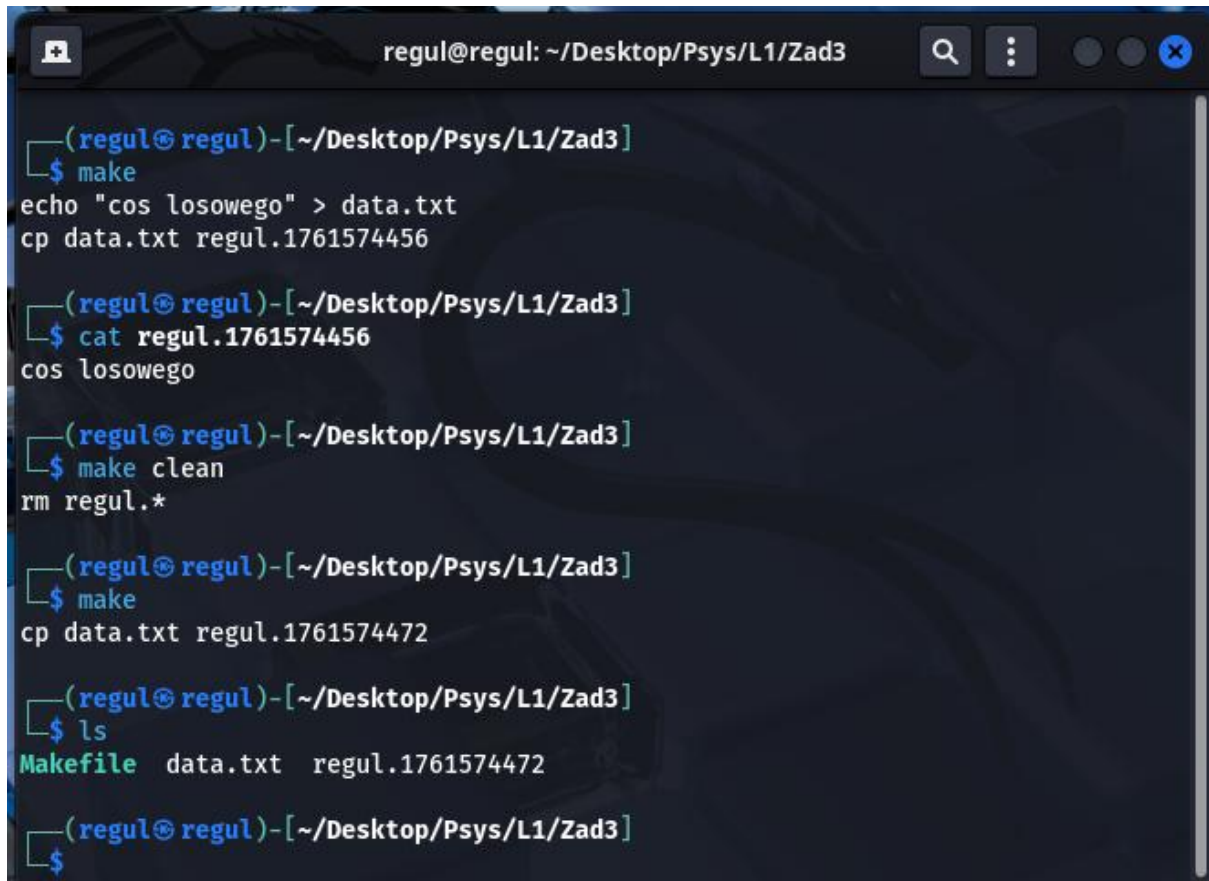
all: data.txt
    cp data.txt $(File)
clean:
    rm $(Username).*

data.txt:
    echo "cos losowego" > data.txt

-- INSERT --
```

Makefile z drugiego zadania

Makefile z drugiego zadania zawiera te same zmienne. Target „clean” też zostaje taki sam, natomiast pojawia się nowy target „data.txt” który odpowiada za utworzenie pliku data.txt lub jego aktualizacje. Target „all” teraz kopiuje zawartość pliku data.txt do „naszego” pliku, target „all” działa zależnie od targetu „data.txt”, czyli przy wykonywaniu „all” zawsze wykona się „data.txt”.

A terminal window with a dark background and light blue text. The window title is 'regul@regul: ~/Desktop/Phys/L1/Zad3'. The terminal shows a series of commands and their outputs. The first command is 'make', which outputs 'echo "cos losowego" > data.txt' and 'cp data.txt regul.1761574456'. The second command is 'cat regul.1761574456', which outputs 'cos losowego'. The third command is 'make clean', which outputs 'rm regul.*'. The fourth command is 'make', which outputs 'cp data.txt regul.1761574472'. The fifth command is 'ls', which outputs 'Makefile data.txt regul.1761574472'. The prompt '\$' is visible at the end of each command line.

```
(regul@regul)-[~/Desktop/Phys/L1/Zad3]
$ make
echo "cos losowego" > data.txt
cp data.txt regul.1761574456

(regul@regul)-[~/Desktop/Phys/L1/Zad3]
$ cat regul.1761574456
cos losowego

(regul@regul)-[~/Desktop/Phys/L1/Zad3]
$ make clean
rm regul.*

(regul@regul)-[~/Desktop/Phys/L1/Zad3]
$ make
cp data.txt regul.1761574472

(regul@regul)-[~/Desktop/Phys/L1/Zad3]
$ ls
Makefile data.txt regul.1761574472

(regul@regul)-[~/Desktop/Phys/L1/Zad3]
$
```

Wykonanie make oraz make clean

Zadanie 3


```
(regul@regul)-[~]  
$ cd /home/regul/Desktop/Psys/L1/Zad1  
  
(regul@regul)-[~/Desktop/Psys/L1/Zad1]  
$ make  
cp data.txt build/regul.1761576063  
  
(regul@regul)-[~/Desktop/Psys/L1/Zad1]  
$ cd /home/regul/Desktop/Psys/L1/Zad1/build  
  
(regul@regul)-[~/.../Psys/L1/Zad1/build]  
$ ls  
regul.1761571116  regul.1761575285  regul.1761576021  
regul.1761575278  regul.1761575986  regul.1761576063  
  
(regul@regul)-[~/.../Psys/L1/Zad1/build]  
$ cd /home/regul/Desktop/Psys/L1/Zad1/  
  
(regul@regul)-[~/Desktop/Psys/L1/Zad1]  
$ make clean  
rm -rf build data.txt  
  
(regul@regul)-[~/Desktop/Psys/L1/Zad1]  
$ is  
s: command not found  
  
(regul@regul)-[~/Desktop/Psys/L1/Zad1]  
$ ls  
Makefile
```