

1.

```
GNU nano 8.3
username=$(shell whoami)
timespan=$(shell date +%s)
filename=$(username)_$(timespan).txt

all:
    touch ${filename}

clean:
    rm ${username}_*.txt
```

username, timespan, filename - to zmienne

whoami - pokazuje nazwę aktualnego użytkownika. polecenie \$(shell whoami) wykonuje się w shellu i zapisuje wynik do zmiennej username

date +%s – to aktualny czas w sekundach. polecenie \$(shell date +%s) wykonuje się w shellu i zapisuje wynik do zmiennej timespan

file łączy wartości dwóch zmiennych i tworzy nazwę pliku

target all tworzy pusty plik, target clean usuwa wszystkie pliki, które pasują do podanego wzoru

wynik działania Makefile:

```
karolina@karolina:~/ProgSys/make/lab2/zad1$ make
touch karolina_1761144845.txt
karolina@karolina:~/ProgSys/make/lab2/zad1$ make clean
rm karolina_*.txt
```

polecenie make utworzyło plik z nazwą składającą się z mojego loginu i czasu

polecenie make clean usunęło plik przy pomocy komendy rm karolina\_\*.txt

2.

```
GNU nano 8.3                                     Makefile
username=$(shell whoami)
timespan=$(shell date +%s)
filename=$(username)_$(timespan).txt

all: data.txt
    cp data.txt $(filename)

data.txt:
    echo "Hello, world!" > data.txt

clean:
    rm ${username}*.txt
```

pojawiła się zależność – target all zależy od data.txt -> przed wykonaniem all, Make sprawdza, czy plik data.txt istnieje, a jeśli nie, to go tworzy(reguła data.txt)

target data.txt tworzy plik data.txt z zawartością „Hello, world!” za pomocą polecenia echo

wynik działania Makefile:

```
karolina@karolina:~/ProgSys/make/lab2/zad1$ make
echo "Hello, world!" > data.txt
cp data.txt karolina_1761145641.txt
```

polecenie make utworzyło plik data.txt, a następnie skopiowało go do pliku z nazwą składającą się z mojego loginu i czasu

3.

```
GNU nano 8.3                                     Makefile
username=$(shell whoami)
timespan=$(shell date +%s)
filename=$(username)_$(timespan).txt
outdir="build"

all: data.txt $(outdir)
    cp data.txt $(outdir)/$(filename)

data.txt:
    echo "Hello, world!" > data.txt

clean:
    rm -rf $(outdir)
    rm data.txt

$(outdir):
    mkdir $(outdir)
```

pojawiła się nowa zmienna outdir – teraz wszystkie pliki kopiują się do folderu build  
target all zależy od dwóch innych targetów: data.txt i \$(outdir). Oznacza to, że Make najpierw tworzy folder i plik, a potem kopiuje data.txt do pliku o nazwie filename

pojawiła się też zależność przez zmienną – \$(outdir) to nie jest konkretna nazwa, tylko zmienna, którą można zmienić. Na przykład, jeśli zmienimy outdir="dir", wszystko dalej będzie działać poprawnie

target clean teraz usuwa cały folder i czyści wszystko

wynik działania Makefile:

```
karolina@karolina:~/ProgSys/make/lab2/zad1$ make
echo "Hello, world!" > data.txt
mkdir "build"
cp data.txt "build"/karolina_1761499956.txt
karolina@karolina:~/ProgSys/make/lab2/zad1$ make clean
rm -rf "build"
rm data.txt
```

polecenie make utworzyło plik data.txt i katalog build, do którego został skopiowany plik data.txt jako karolina\_1761499956.txt, make clean usunęło katalog i plik data.txt

Potwierdzenie działania po zmianie outdir="dir":

```
username=$(shell whoami)
timespan=$(shell date +%s)
filename=$(username)_$(timespan).txt
outdir="dir"

all: data.txt $(outdir)
    cp data.txt $(outdir)/$(filename)

data.txt:
    echo "Hello, world!" > data.txt

clean:
    rm -rf $(outdir)
    rm data.txt

$(outdir):
    mkdir $(outdir)
```

```
karolina@karolina:~/ProgSys/make/lab2/zad1$ make
echo "Hello, world!" > data.txt
mkdir "dir"
cp data.txt "dir"/karolina_1761503898.txt
```