

```
target=hello
filename=hello.c
fileO=hello.o
outdir=build
dir=src
cc=gcc
cflags=-Wall -g

all: $(target)

$(target): $(outdir) $(fileO)
    $(cc) $(outdir)/$(fileO) -o $(outdir)/$(target)
    cp $(dir)/$(filename) $(outdir)/$(filename)

$(fileO): $(dir) $(outdir)
    $(cc) -c $(dir)/$(filename) -o $(outdir)/$(fileO)

clean:
    rm -rf $(outdir)

$(outdir):
    mkdir -p $(outdir)
```

target, filename, fileO, outdir, dir, cc, ccglags - to zmienne

target – nazwa programu wynikowego

filename – nazwa pliku źródłowego

fileO - nazwa pliku o, wynik kompilacji

outdir, dir - katalogi

cc – kompilator języka C

ccglags – flagi kompilacji

target \$(target) tworzy program wynikowy hello. zależności : katalog build oraz plik hello.o

gcc \$(outdir)/\$(fileO) -o \$(outdir)/\$(target) kompiluje plik

cp kopiuje plik źródłowy do katalogu build

target \$(fileO) tworzy plik hello.o z pliku hello.c. zależności : katalog src i katalog build

-c – kompilacja bez linkowania; -o określa, gdzie ma być zapisany wynik

target clean usuwa katalog build oraz wszystkie pliki w nim zawarte

target \$(outdir) tworzy katalog

wynik działania Makefile:

```
karolina@karolina:~/ProgSys/make/lab3$ make
mkdir -p build
gcc -c src/hello.c -o build/hello.o
gcc build/hello.o -o build/hello
cp src/hello.c build/hello.c
karolina@karolina:~/ProgSys/make/lab3$ rm build/hello
karolina@karolina:~/ProgSys/make/lab3$ make
gcc -c src/hello.c -o build/hello.o
gcc build/hello.o -o build/hello
cp src/hello.c build/hello.c
karolina@karolina:~/ProgSys/make/lab3$ ls
build  hello  hello.o  Makefile  src
```

polecenie make sprawdza, czy istnieje katalog build, jeśli nie, tworzy go; kompiluje plik hello.c, następnie tworzy plik wykonywalny hello, kopiuje hello.c do katalogu build

aby sprawdzić poprawność działania pliku Makefile, usunęłam ręcznie hello, a następnie ponownie uruchomiłam polecenie make. po wykonaniu tej komendy program został ponownie skompilowany i uruchomiony bez żadnych błędów.