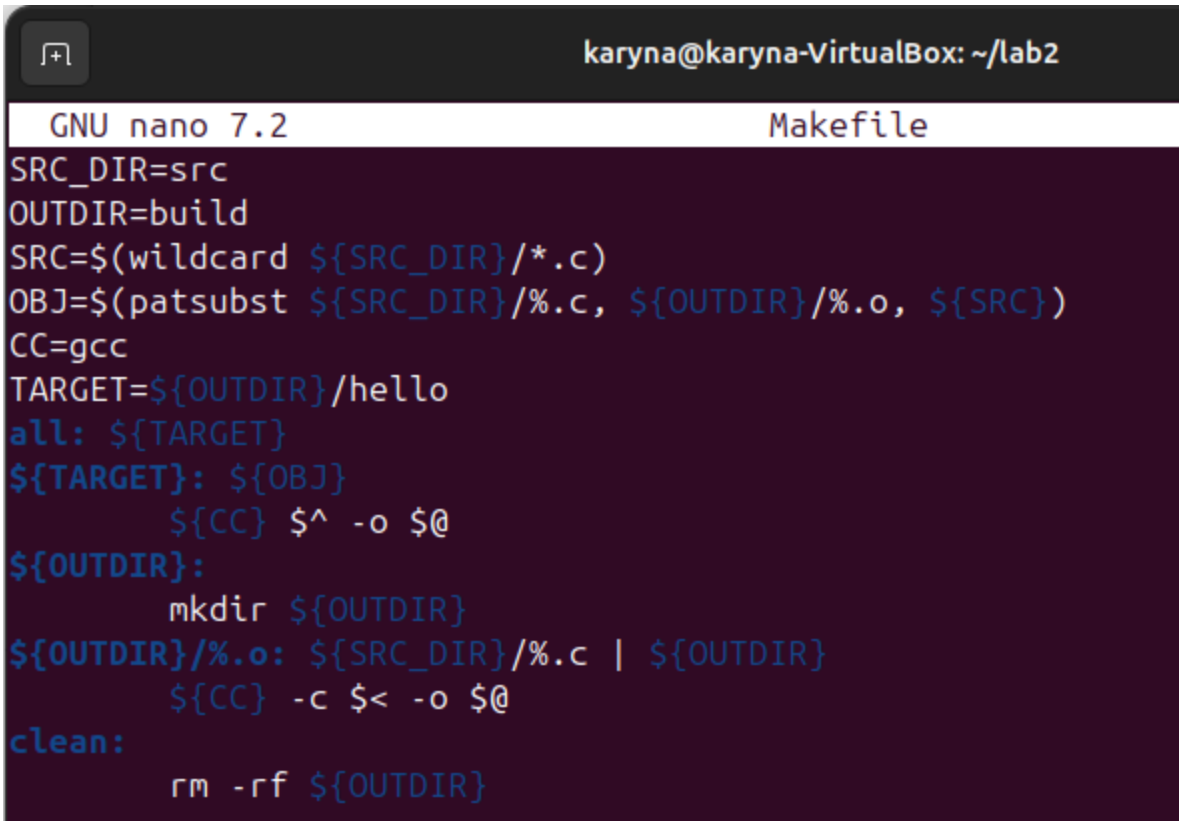

Zmienne takie jak SRC_DIR, OUTDIR, CC ułatwiają wprowadzanie zmian w Makefile, po zmianie jednej zmiennej cały kod nadal działa prawidłowo.

- **SRC=$(wildcard ${SRC_DIR}/*.c)**

Zbiera listę wszystkich plików *.c w katalogu src i zapisuje je w zmiennej SRC.

Wildcard to funkcja w Makefile, która zbiera tylko te pliki, które pasują do wzoru (np. wszystkie pliki .c w katalogu).

- **OBJ=$(patsubst ${SRC_DIR}/%.c, ${OUTDIR}/%.o, ${SRC})**

Zamienia nazwy plików źródłowych na odpowiadające pliki obiektowe w build. Rezultat zapisuje w OBJ.

np. src/plik.c -> build/plik.o

Funkcja patsubst zbiera pliki, które kończą się na .c i zamienia je na pliki .o. Zmienna SRC zawiera listę plików źródłowych .c, dla których wykonujemy zmianę (.o).

* szuka pliki, pasujące do wzórca, np. .c lub .o

% dopasowuje pliki o tej samej nazwie przed kropką (czyli nazwy plików przed rozszerzeniem .c muszą być takie same)

- **TARGET=${OUTDIR}/hello**

Pełna ścieżka i nazwa programu wynikowego. W naszym wypadku: build/hello.

- **all: ${TARGET}**

Domyślny cel. Jest zależny od ${TARGET}.

- **${TARGET}: ${OBJ} zależny od OBJ**

Do stworzenia pliku wynikowego (build/hello) potrzebujemy wszystkich plików obiektowych wymienionych w ${OBJ}.

- **${CC} $^ -o $@**
- **$^ wszystkie zależności**
- **$@ to jest skrót na target**
  uruchamia gcc; -o $@ zapisuje wynik w pliku, który jest celem ($@ = target); $^ oznacza wszystkie zależności, czyli wszystkie pliki .o
- **${OUTDIR}:**
- mkdir tworzy katalog build, jeśli nie istnieje.
- **${OUTDIR}/%.o: ${SRC_DIR}/%.c | ${OUTDIR}**
  Tworzymy pliki obiektowe (.o) w katalogu build, zależności: bierzemy pliki źródłowe (.c) z katalogu src.
- | przed skompilowaniem pliku `.c` do pliku `.o`, upewniamy się, że katalog `build` istnieje.
- jeśli wszystkie pliki są już zbudowane i nic się nie zmieniło,przy ponownym uruchomieniu wypisze komunikat:

```
karyna@karyna-VirtualBox:~/lab2$ make
make: Nothing to be done for 'all'.
```

- **${CC} -c $< -o $@: kompilacja**
- **$< to pierwsza zależność**
  -c kompilujemy plik źródłowy ($<, czyli ${SRC_DIR}/%.c) i –o zapisujemy do pliku obiektowego $@, ${OUTDIR}/%.o
- **clean:**
  rm -rf ${OUTDIR}

Usuwa katalog build i wszystkie jego pliki.

Wynik:

```
karyna@karyna-VirtualBox:~/lab2$ make
mkdir build
gcc -c src/hello.c -o build/hello.o
gcc -c src/main.c -o build/main.o
gcc build/hello.o build/main.o -o build/hello
karyna@karyna-VirtualBox:~/lab2$ make
make: Nothing to be done for 'all'.
karyna@karyna-VirtualBox:~/lab2$ make clean
rm -rf build
```