

Programowanie systemowe

Sprawozdanie

Aron Krajda - 283874

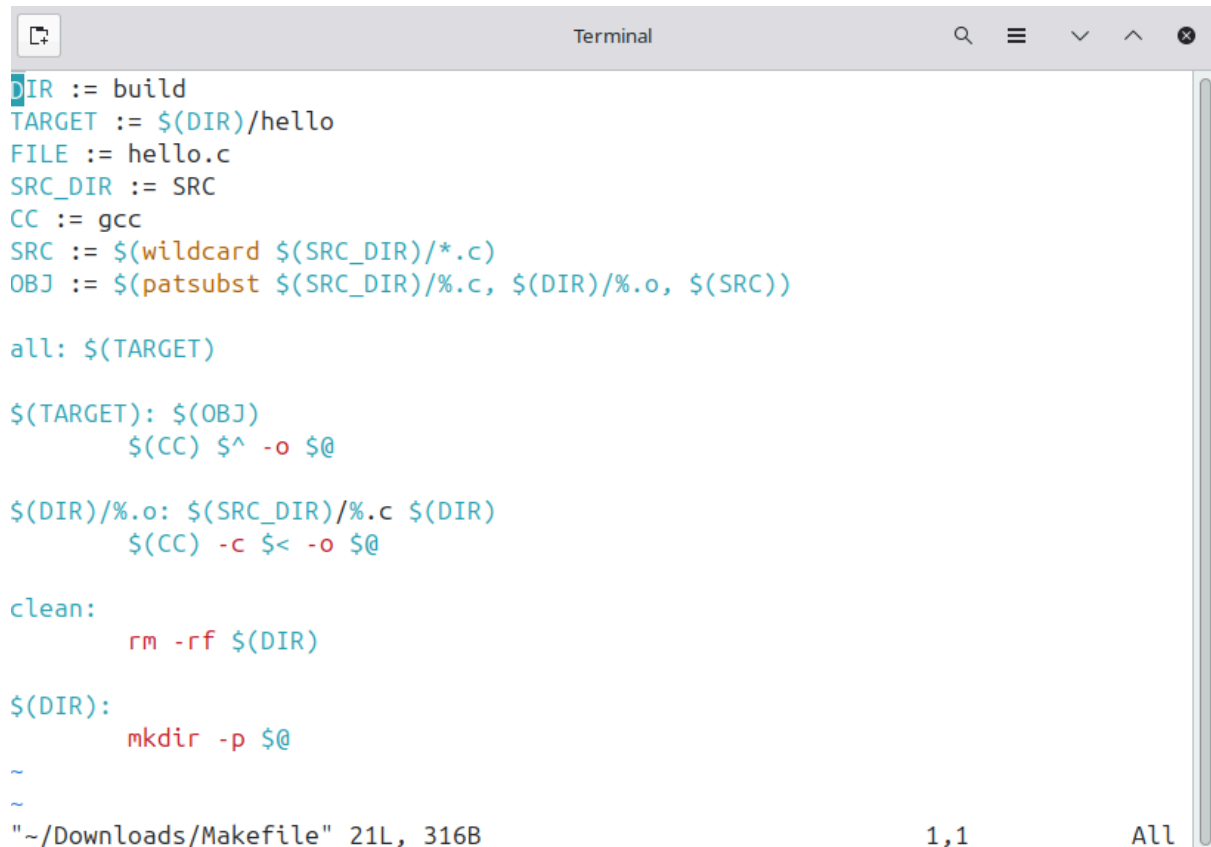
Data wykonania sprawozdania: 06.11.2025

Cele:

Celem zadania było rozwinięcie poprzedniego pliku i wprowadzenie w nim nowo poznanych funkcji [m.in.](#) kompilowania wszystkich plików z rozwinięciem .c w danym folderze, czy dodanie automatycznych zmiennych.

Pliki:

Plik Makefile:



```
IR := build
TARGET := $(DIR)/hello
FILE := hello.c
SRC_DIR := SRC
CC := gcc
SRC := $(wildcard $(SRC_DIR)/*.c)
OBJ := $(patsubst $(SRC_DIR)/%.c, $(DIR)/%.o, $(SRC))

all: $(TARGET)

$(TARGET): $(OBJ)
    $(CC) $^ -o $@

$(DIR)/%.o: $(SRC_DIR)/%.c $(DIR)
    $(CC) -c $< -o $@

clean:
    rm -rf $(DIR)

$(DIR):
    mkdir -p $@

~
~
~/Downloads/Makefile" 21L, 316B 1,1 All
```

Do utworzonego na poprzednich laboratoriach pliku Makefile dodano dwie nowe zmienne: SRC oraz OBJ. Zmienna SRC zawiera wykrywanie wszystkich plików kończących się na .c, a zmienna OBJ zawiera tworzenie odpowiadających im plików z rozszerzeniem .o. Powoduje to, że możliwe jest wykorzystanie pliku Makefile do kompilacji wielu plików napisanych w C jednocześnie.

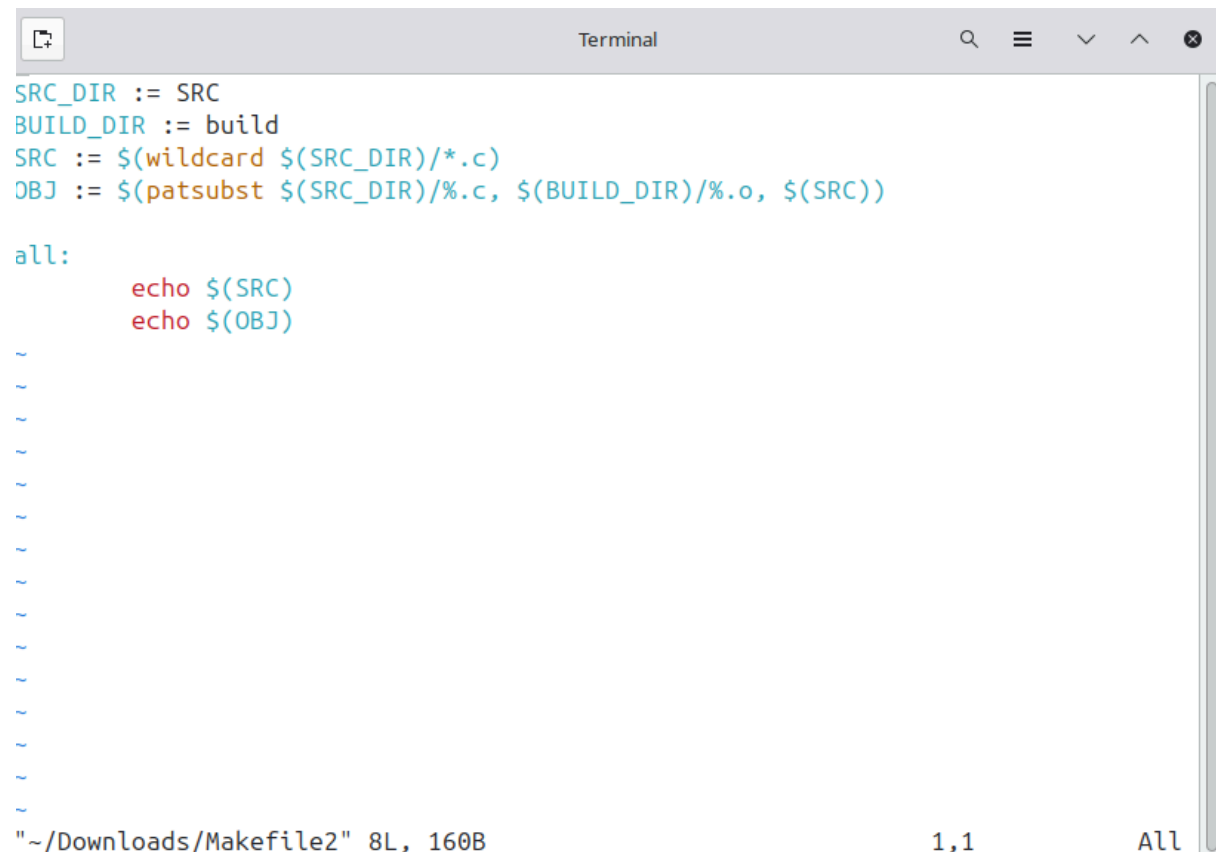
Miedzy regułą \$(TARGET) i regułą clean dodano regułę, która kompiluje pliki .c do plików .o.

Dodatkowo w możliwych miejscach dodano zmienne automatyczne, czyli odwoływanie się do zmiennych po lewej i prawej stronie ":" w regułach zamiast do zmiennych po ich nazwie. Powoduje to, że łatwiej zmienić nazwy zmiennych lub zmienne, do których chcemy się odwołać.

Usunięto też odwołania do konkretnych plików lub katalogów i w każdym miejscu odwołanie jest do zmiennej zamiast konkretnych plików/katalogów.

Domyślny cel all, podobnie jak w przypadku pliku Makefile stworzonego na poprzednich laboratoriach kompiluje - tym razem wszystkie znajdujące się w katalogu SRC pliki napisane w języku C, wcześniej dokonując kompilacji bez linkowania.

Plik Makefile2:

A screenshot of a terminal window titled "Terminal". The window displays the contents of a file named "Makefile2". The text is as follows:

```
SRC_DIR := SRC
BUILD_DIR := build
SRC := $(wildcard $(SRC_DIR)/*.c)
OBJ := $(patsubst $(SRC_DIR)/%.c, $(BUILD_DIR)/%.o, $(SRC))

all:
    echo $(SRC)
    echo $(OBJ)
```

The terminal shows a prompt character "~" on several lines. At the bottom of the terminal, it shows the file path "~/Downloads/Makefile2" with a size of 8L, 160B. On the right side, it shows "1,1" and "All".

Stworzono również plik Makefile2, który miał za zadanie testować działania komend wildcard oraz patsubst. Domyślny cel all wywoływany komendą make służy do wyświetlania plików w katalogu SRC, które kończą się na .c oraz wyświetlenia stworzonych odpowiadających im plików z rozszerzeniem.o.

Podsumowanie:

Oba pliki - zarówno Makefile jak i Makefile2 działają poprawnie. Plik Makefile po wpisaniu komendy make kompiluje pliki z rozszerzeniem .c znajdujące się w katalogu SRC. Plik Makefile2 poprawnie testuje działania komend wildcard i patsubst i pozwala zapoznać się z działaniem tych komend. Użycie automatycznych zmiennych pozwala uprościć wprowadzanie zmian w pliku, a użycie komend wildcard i patsubst pozwala na to, aby plik

stał się bardziej skalowalny i był w stanie obsługiwać więcej niż 1 plik w języku C jednocześnie.